

RESEARCH ARTICLE

The Built Environment Data Framework for Simulated Design and Vulnerability Modelling in Earthquake Engineering

Volkan Ozsarac^{1,2}  | Nuno Pereira³  | Hossameldeen Mohamed⁴  | Xavier Romão³  | Gerard J. O'Reilly^{1,2} 

¹European Centre for Training and Research in Earthquake Engineering, Pavia, Italy | ²Centre for Training and Research on Reduction of Seismic Risk, Scuola Universitaria Superiore IUSS, Pavia, Italy | ³Faculty of Engineering, University of Porto, Porto, Portugal | ⁴Faculty of Engineering, Aswan University, Aswan, Egypt

Correspondence: Gerard J. O'Reilly (gerard.oreilly@iusspavia.it)

Received: 12 February 2025 | **Revised:** 31 March 2025 | **Accepted:** 7 May 2025

Funding: This study was carried out with financial support from the European Union through the EPOS-ON (Grant Agreement No. 101131592) project and the financial support of Funding—UID/04708 of the CONSTRUCT—Instituto de I&D em Estruturas e Construções—funded by Fundação para a Ciência e a Tecnologia, I.P./MCTES through the national funds.

Keywords: building portfolio | building variability | design codes and regulations | nonlinear model | open-source | seismic design

ABSTRACT

The seismic vulnerability modelling of different categories of buildings analytically requires methodologies capable of capturing the wide range of building standards, construction practices, architectural layouts, earthquake design scenarios and available knowledge. Current vulnerability models employ varying assessment approaches, building taxonomies, representations of seismic loading and, in some cases, rely on a limited number of archetype structural models to represent an entire building class. Consequently, these structural models are likely to oversimplify the seismic behaviour of an individual building, fail to adequately capture the reality of building-to-building variability and inadequately account for multiple sources of uncertainty, particularly when applied to regional contexts. Addressing these issues requires a probabilistic approach where seismic vulnerability is assessed using models of building portfolios that can reflect features related to engineering design practice, as well as construction variability and quality. To achieve this objective, this article introduces a collaborative framework for the simulated design of buildings, along with the open-source software package developed for its integration into the Built Environment Data initiative, a planned service within the European Plate Observing System. The simulated design framework accommodates the design of buildings using both historical and modern seismic design procedures and regulations, while capturing building-to-building variability. Following the design process, the framework generates OpenSees computational models for nonlinear analysis, facilitating the development of probabilistic seismic demand models, fragility functions and vulnerability models. The framework's capabilities are demonstrated through examples that highlight notable distinctions among the building classes under consideration and emphasise the importance of the attributes involved in the design process. The collaborative nature of the framework presented here enables the earthquake engineering community to contribute to a growing database of seismic design practices, encompassing a wide range of design codes.

1 | Introduction

Natural hazards such as earthquakes, floods and hurricanes pose significant threats to human life, infrastructure and economic stability. While the occurrence of these events (hazard assessment) and the presence of people and assets in affected areas (exposure assessment) are important, understanding how these elements respond to hazard events (vulnerability assessment) is crucial for an accurate risk evaluation. Vulnerability modelling, therefore, plays a critical role in comprehensive risk assessments for earthquakes and other natural hazards. It involves evaluating the susceptibility of structures, populations and systems to damage when exposed to hazard events. By considering factors such as building design, construction materials, maintenance practices and socio-economic conditions, physical vulnerability models help predict the extent of damage in the built environment and inform the development of effective risk mitigation strategies [1].

In seismic risk assessment, physical vulnerability modelling focuses on the response of buildings and infrastructure to earthquake-induced ground shaking. This involves developing fragility functions, which describe the probability of a structure reaching or exceeding a certain damage state, given the occurrence of a specific intensity of ground shaking. Over time, different methods have been employed to develop these functions, including analytical, empirical and hybrid methods [2]. Among these, analytical methods (e.g., [3, 4]) are advantageous because they provide a quantitative assessment of structural behaviour without relying on expert judgment, thus minimising the subjectivity perceived in other approaches. Furthermore, analytically derived fragility functions provide a transparent and thorough framework for managing uncertainties. However, the assumptions and idealisations inherent to the selected analytical modelling approach can limit their ability to fully capture the complexities of actual structural behaviour, raising questions about the reliability of the derived fragility functions when compared to empirically collected data (see [5]). In addition, these methods can be computationally demanding and resource-intensive, making them less practical in earlier studies when computational power was limited.

As such, various approaches have been employed worldwide to develop fragility functions through independent research, regional initiatives and government-funded programmes. In the United States, with support from the Federal Emergency Management Agency, numerous fragility functions representing common building typologies have been derived and implemented in the HAZUS software package [6]. In Europe, several European Union-funded projects, such as the RISK-UE (2001–2004) [7] and LESSLOSS (2004–2007) [8], have contributed significantly to the development of fragility and vulnerability functions. In South America, Villar-Vega et al. [9], in collaboration with local experts, developed fragility functions for different building classes. Moreover, fragility functions have been developed for the dominant building typologies in specific countries by various researchers, including Erberik [10, 11] and Erdik et al. [12] for Türkiye, Borzi et al. [13, 14] for Italy, Salgado et al. [15] for Colombia, Silva et al. [16, 17] for Portugal and Motamed et al. [18] for Iran, for example. Among these studies, different approaches were adopted to represent the structural system

(i.e., simplified single- or multi-degree-of-freedom models, i.e., SDOF or MDOF models), and the potential variability of seismic design regulations over time was not always captured with the same level of detail. While these studies have significantly advanced the field of seismic vulnerability modelling, they also highlight several challenges and limitations, especially for large-scale risk assessment applications. In this context, a common issue is the generalisation of models, which often rely on diverse assessment approaches, broad building class definitions, and, in some instances, a limited number of representative structures to characterise an entire building class. Consequently, existing models frequently fail to fully capture building-to-building variability and to adequately account for multiple sources of uncertainty. Moreover, it is seen that analytical vulnerability assessment strategies vary between two extremes—the use of many simplified SDOF models to capture inter-building variability, or a few detailed MDOF models to explore intra-building uncertainty, with computational power typically being the limiting factor. As highlighted by Silva et al. [19], both approaches involve trade-offs in uncertainty treatment and scalability, suggesting that an optimal solution may lie in combining the two, for example, through the calibration of simpler SDOF models using detailed MDOF models. Addressing these challenges requires a more systematic approach that accounts for the wide range of construction practices, building codes, architectural layouts, seismic design scenarios, and material quality across regions while supporting the transition between two vulnerability modelling approaches.

To address some of these issues, the recent European exposure model [20] developed within the scope of the Horizon 2020 project SERA (<http://www.sera-eu.org>) introduced the evolution of seismic design codes and the seismic demand zonation as attributes of the building class taxonomy for reinforced concrete (RC) buildings. The idea behind the proposed taxonomy consisted of complementing the morphological data that is often available (either based on existing national statistical data [21, 22], sidewalk surveys [23], image processing [24], or remote-sensing techniques [25]) with information about the expected strength and ductility of these buildings. As shown by Crowley et al. [20], when creating an exposure model to be used at a continental scale, only a few morphological attributes of buildings, such as the number of storeys, the year of construction, the main construction material and the type of structural system are usually possible to be adopted, based on existing data. Previous studies have also used ductility as an attribute aiming to distinguish different building classes, by adopting mapping schemes [26, 27] that combine the year of construction and/or the seismicity of the region to establish possible ductility classes. Nonetheless, such definitions can be limited in terms of flexibility (in time and geography) and are not likely to allow the use of the same vulnerability model at a continental scale or local level. A major step towards achieving this flexibility was made by Crowley et al. [28] by proposing a mapping scheme that decouples the seismic strength (therein represented by the lateral load coefficient, β , defined as a percentage of the building weight) from the seismic design principles (namely their respective ductility-related aspects). In that mapping scheme, the taxonomy is comprised of four design classes, representing the prevalent seismic design practices in Europe during different periods (CDN: absence of seismic design, CDL: designed for lateral resistance using allowable stress design,

CDM: designed for lateral resistance with modern limit state design, and CDH: designed for lateral resistance as in CDM coupled with target ductility requirements). The adoption of these categories provided a harmonised classification of seismic design codes and approaches across Europe, which reflect not only major changes in the seismic zonation but also the variations in seismic design provisions, structural design principles and construction practices over time. Notably, it is envisioned that these efforts focused on RC frame structures in Europe can be further extended to encompass additional building taxonomies within specific country contexts.

Although the approach introduced by Crowley et al. [28] enhances the characterisation of RC buildings, many other attributes, such as geometrical and structural details, are still missing in currently available exposure models. In this regard, simulated design procedures can serve as by-pass strategies to overcome the limited level of information, that is, often available regarding the details of a building design, as recognised by the European seismic safety assessment standards, that is, the Eurocode 8–Part 3 [29], for assessing the seismic safety of an RC building where there is limited knowledge. By simulating the design procedure, that is, what an engineer during that period would have likely done, only a few geometric variables, material properties and construction quality levels are required to generate a possible structural configuration for a building to be assessed. Past research (e.g., [14, 30–39]) has also used simulated design approaches to reduce epistemic uncertainty in the quantification of seismic vulnerability of building portfolios, mostly to automate the generation of building layouts and their consequent design to generate larger samples of building models for seismic performance analysis. In line with this, the seismic vulnerability models of low- to mid-rise RC frame buildings for the 2020 European Seismic Risk Model [40] were developed using a simulated design approach involving randomised building portfolios. In particular, to account for the building-to-building variability within a given building class, the geometric variables required for the simulated design procedure were randomly generated based on existing statistical distributions. The resulting designs were then idealised as nonlinear SDOF systems and analysed using a record-to-record uncertainty propagation method (i.e., cloud analysis) to develop fragility and vulnerability functions.

Despite these advancements, it is important to highlight that the statistical information about the general building properties, such as span lengths, storey heights and material strength properties, can vary significantly depending on the country or region. Moreover, the implementation of building codes and design practices is different in each country, even today. In fact, a detailed analysis of a specific country's standards (e.g., [41]) can reveal deviations from the generalised building class definitions proposed in Crowley et al. [20], especially when seismic events have led to changes in the country's design practices. While existing simulated design procedures typically align with the seismic design codes in effect at the time, their focus has remained on country-specific applications and localised studies, and no generalised framework has yet been developed to address these challenges on a broader scale. Moreover, existing procedures usually do not incorporate certain common design practices (e.g., column uniformity over height, preferred section dimensions and reinforcement ratio limits) that can influence seismic vul-

nerability. Another key limitation lies in the lack of conformity between the actual buildings and those generated strictly by following design rules. Construction quality can affect material properties (i.e., mean values and variability), construction details (i.e., stirrup spacing, confinement, anchorage effectiveness) and geometrical variables, but including these issues into seismic design frameworks is rarely implemented. Numerical modelling choices influenced by the construction quality are often absent from traditional vulnerability models, creating further gaps in their ability to provide realistic risk assessment results compared to the observed reality following seismic events [5]. Furthermore, there has been no effort to develop an integrated simulated design-based framework that allows analysts to incorporate modelling strategies capable of addressing building class-specific deficiencies identified through experimental (e.g., [42, 43]) or reconnaissance studies (see [44–47]).

To address these challenges systematically, this article presents a flexible and unified framework for the simulated design of buildings. The proposed framework builds on the developments of the SERA project, combined with advancements in object-oriented programming implemented in Python. The framework's key innovation lies in employing the principle of composition over inheritance to derive design solutions for individual structural components and the structure as a whole, while preserving the generality and adaptability of the design procedures. Moreover, the randomised building portfolios produced by the simulated design are then used to generate computational models that can account for construction quality factors, thus facilitating the development of vulnerability models that can effectively capture building-to-building variability. This article presents the main components of the proposed framework for the case of RC frame buildings, but the same approach is extendable to other building typologies. The framework's capabilities are illustrated based on examples of European design practices that showcase relevant differences among building types and emphasise the critical role of design and structural attributes in the vulnerability assessment. Finally, it is highlighted that, by using open-source software implemented in programming languages commonly taught in university engineering programmes, the structural engineering community is encouraged to contribute to expanding the proposed framework, namely by introducing components that reflect other seismic design practices included in past or current building design codes. This collaborative environment will be further promoted by integrating this simulated design framework into the Built Environment Data (BED) initiative (<http://builtenvdata.eu>), a forthcoming service within the European Plate Observing System (EPOS). The BED initiative aims to provide access to data and services related to the built environment, with recent examples being the *Experiments* database outlined by Shah-nazaryan et al. [48] offering access to experimental data from different laboratories, or the *Embodied Carbon* service developed by Caruso et al. [49]. The *SimDesign* framework presented here therefore integrates with this broader goal of BED and is available at <https://simdesign.builtenvdata.eu>.

2 | Framework Overview

The proposed simulated design framework provides a clear workflow for designing structures that represent typical buildings

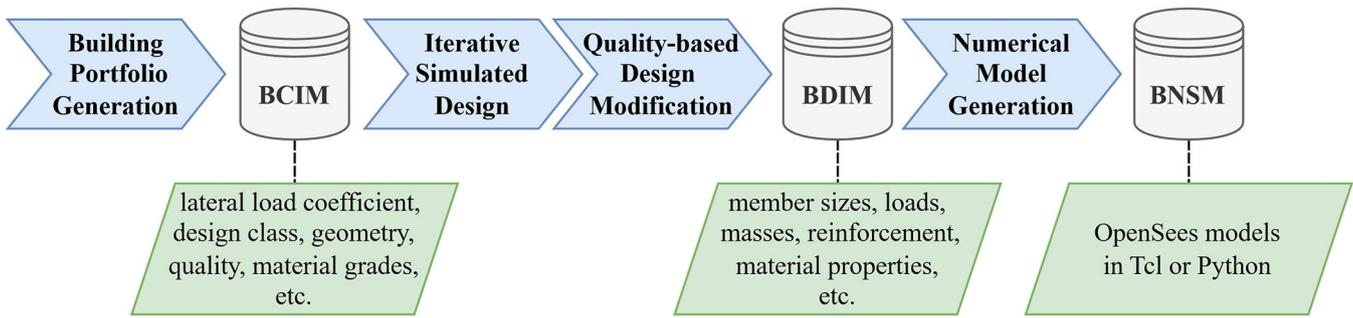


FIGURE 1 | General overview of the workflow defined for the proposed simulated design framework.

of a country or region, and generating the corresponding 3D nonlinear numerical models in OpenSees [50]. These models can ultimately support the generation of vulnerability models where the key variables affecting the building behaviour are clearly known, and thus, the building taxonomy can be well-described. To this end, the proposed framework follows the workflow involving the four main steps illustrated in Figure 1. It is noted the framework developed and described herein is for the case of RC moment resisting frames (MRFs), but the same approach is envisaged to be extended to other structural typologies (e.g., braced frames, shear wall buildings) and materials (e.g., steel, precast).

The first step involves generating an information dataset, which captures the general characteristics of buildings constructed, or planned for construction, in a specific area, to guide the design of buildings within a portfolio. This dataset includes primary attributes that define the selected building class, remain consistent across the building portfolio and serve as inputs to the framework along with the size of the portfolio. These attributes are, namely, the number of storeys, the lateral load coefficient, β , and the design class, which represents the regional or country-specific seismic design practices adopted during a certain period of time (e.g., CDL as defined by [28]). Additionally, the dataset contains secondary attributes, which involve information such as typical material grades and the expected construction quality level, as well as geometry variables such as in-plan configurations and storey heights. Collectively, the secondary attributes and the geometry variables account for building-to-building variability and are established through random sampling from probability distributions specific to a given design class. While these probability distributions are often readily available (e.g., [16, 51, 52]), they can also be derived through collaborative efforts, and adjusted to reflect the regional or country-specific contexts. At the end of this step, the generated dataset for the building portfolio, or the building realisations, is stored in the *Building Class Information Model* (BCIM) database.

In the second step, each building realisation in the BCIM undergoes a simulated design procedure that aims to replicate the approach an engineer would follow to define a feasible design solution. This process accounts for the regional seismicity, represented by β , and iteratively determines the structural member dimensions and reinforcing steel configurations that meet all the design code requirements and design practice rules. The procedure is adaptable, allowing the incorporation of rules and practices tailored to specific design classes. For instance,

the design can be based on gravity load combinations only, as in the case of older buildings, or expanded to include seismic load combinations. Moreover, as more research on the historical evolution of national design codes advances (e.g., [41]), it will become possible to integrate country-specific design classes while preserving the generality of the iterative design algorithms. This adaptability fosters the framework's continuous development, making it relevant across diverse regions and evolving seismic design practices. In the next step, a quality-based modification of the designed structure is introduced to account for construction quality and potential spatial irregularities. More specifically, for each structural component, deviations in material properties and reinforcement detailing from expected values are introduced based on the construction quality level—categorised qualitatively as *Low*, *Moderate* or *High*. At the end of this stage, the final design details of each building, such as material properties, reinforcement configurations and section dimensions, are stored in the *Building Design Information Model* (BDIM) database.

In the final step, the building design data are used to develop 3D nonlinear numerical models in OpenSees [50], utilising both the traditional .tcl interpreter and the recent Python .py interpreter [53]. These numerical models incorporate structural features associated with both the construction quality and the design class. For instance, shear failure of columns is modelled when capacity design principles are not applied for a specific design class, and bond-slip effects are included to account for poor construction quality. In addition to the numerical models, routines for performing modal and nonlinear static pushover analyses of each building are saved and stored in the *Building Nonlinear Structural Model* (BNSM) database.

To enable the described level of control, the framework was implemented in Python using object-oriented programming [54]. While the current implementation only includes RC MRFs, the workflow is adaptable to buildings with other structural systems. In order to ensure such modularity and scalability for future extensions, the proposed *rcmrf* framework is integrated within the broader umbrella library, *simdesign* (available at <https://github.com/builtenvdata/simulated-design>), as represented in Figure 2. The framework is organised into four distinct sub-packages, *geometry*, *bcim*, *bdim* and *bnsim*, each contributing to the implementation of the workflow described earlier. Among these, the *geometry* package, as the name suggests, is responsible for creating the geometrical representations of buildings, which serve as a foundation for the application of the workflow. The *bcim* package is used for generating the aforementioned BCIM

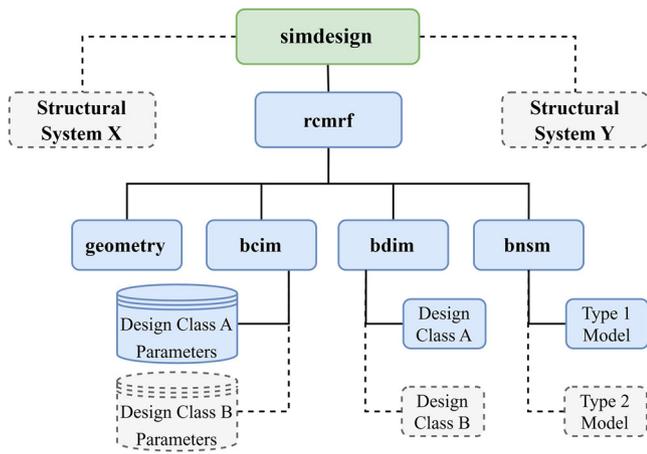


FIGURE 2 | The structure of the simulated design library and the RC MRF design framework.

data in the first step, by utilising a database of decision trees and probability distributions that can be expanded to include additional design classes. The *bcim* package facilitates the generation of BDIM data in the subsequent steps. It dynamically manages various building design classes, where each is implemented in dedicated modules that provide specific components and functionalities, while inheriting general-purpose features from a shared base module. This high level of abstraction allows the seamless integration of new design classes without compromising generality. Finally, the *bnsm* package is utilised to complete the workflow by generating nonlinear bare frame models using a lumped-plasticity approach. Nevertheless, with a similar abstraction, it can be easily modified and extended to incorporate alternative modelling considerations, such as distributed plasticity or infills.

The current Python implementation for RC MRF's includes procedures representative of the generalised European design classes proposed in Crowley et al. [20], but the integration of design procedures for several country-specific design classes is ongoing. In particular, the CDN design class refers to older structures (constructed before the 1960s), designed only for gravity loads using the allowable stress method and typically employing low-strength materials. The design rules from the early standard RBA-1935 [55] are adopted herein for this design class, which are considered to be representative of the European practice at the time. As mentioned in the standard, its design rules were developed based on those found in regulations, standards and guidelines from countries such as Austria, Belgium, Denmark, Germany, Hungary, Italy, Sweden and Switzerland published between 1924 and 1933. The CDL design class considers buildings constructed between the 1960s and 1970s, designed using design codes that introduced seismic design rules involving lateral loads with a distribution pattern proportional to the floor weights. More specifically, structural designs follow the material-specific provisions found in REBA-1967 [56], which are based on allowable stress design or a stress-block approach [57]. As mentioned in the standard, its design rules were developed based on the guidelines published in 1963 by the Comité Européen du Béton (CEB) [58]. Therefore, this standard is also assumed to be generally representative of the European practice at the time. The CDM design class is representative of buildings constructed between

the 1970s and 2000s, designed with the concepts of ultimate strength and partial safety factors, commonly referred to as limit state design, and incorporating improved detailing rules to enhance global ductility, in line with more modern design standards. In this case, the implementation follows the provisions of REBAP-1983 [59, 60], which distribute seismic lateral loads as a function of both floor weights and storey heights. As mentioned in this standard, its design rules were developed based on the 2-volume recommendations published in 1978 by the CEB [61, 62] which include the 1978 Model-Code and influenced the evolution of reinforced concrete design regulations across Europe. Therefore, REBAP-1983 is also assumed to be generally representative of the European practice at the time. Lastly, the CDH design class reflects contemporary seismic design practices (from the early 2000s until now) that implement capacity design principles and reinforcement detailing aimed at achieving specific levels of ductility. More specifically, the design requirements and rules outlined in the Eurocodes [63, 64] for ductility class medium (which is assumed to reflect the most frequently adopted ductility class in Europe) are considered within the implementation.

The following sections address the detailed implementation of the proposed framework, focusing on the generation of the BCIM and BDIM datasets and the development of numerical models for the BNSM. These descriptions aim to establish a solid foundation for future extensions of the framework, therefore ensuring its adaptability and scalability.

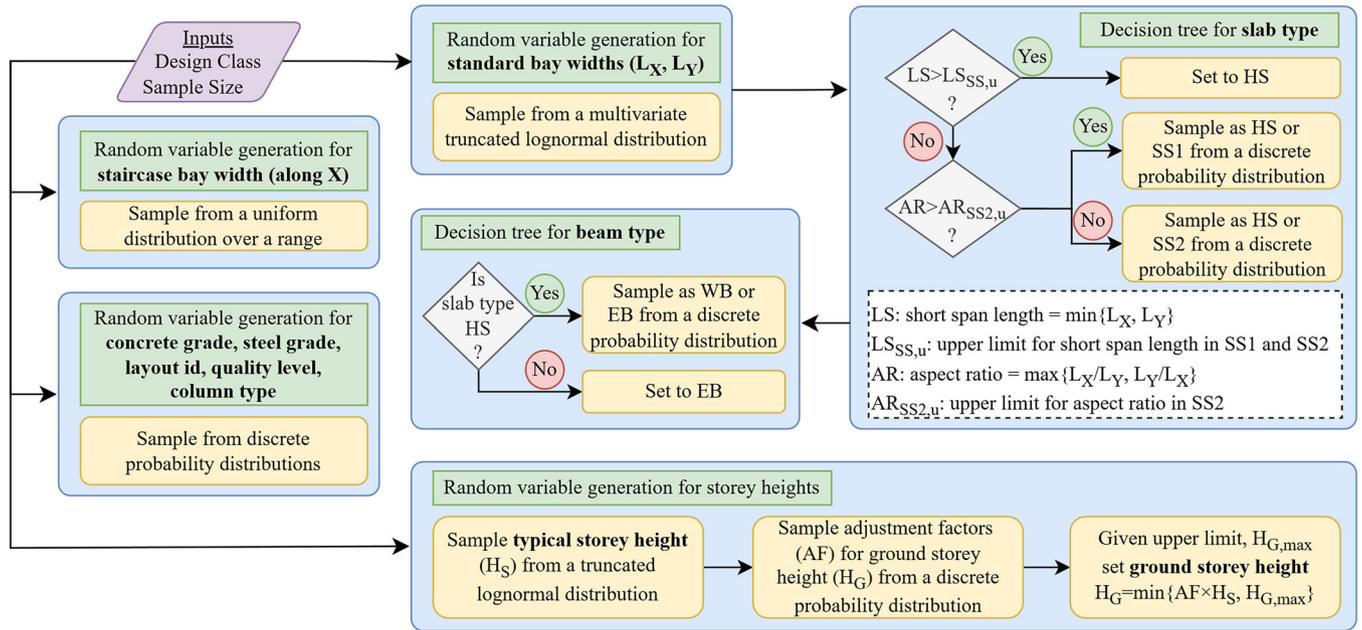
3 | Building Class Information Model

For large-scale seismic risk assessment applications, buildings comprising an exposure model can be classified based on several structural attributes [27]. For instance, the primary attributes considered in the SERA project include the construction material, the lateral load-resisting system, the number of storeys and the expected ductility level, which can be inferred from the seismic design practices in effect at the time of construction (i.e., the design class) and the hazard level (defined by β). These attributes are usually combined to form a taxonomy string [27, 65] to identify what is referred to herein as a building class. When additional information is available, this taxonomy string can be further refined to include information about attributes such as beam and column types, the construction quality, or multi-level attributes like material properties. However, as detailed information is usually unavailable, the uncertainty associated to these secondary attributes should be incorporated into the building portfolio. Moreover, buildings classified with the same taxonomy string within a portfolio often exhibit diverse geometries, meaning they can have different in-plan configurations, bay widths and storey heights. These geometric characteristics are intrinsic to the building portfolio and must also be accounted for to effectively capture building-to-building variability.

In light of this, to incorporate the aforementioned sources of uncertainty when generating a building portfolio, the framework takes the primary attributes (i.e., number of storeys, design class and β) that define the building class as an input and generates samples of the secondary attributes and samples of

TABLE 1 | The secondary taxonomy attributes in BCIM concerning beams, columns, slabs and quality.

Column type	Beam type	Slab type	Construction quality
Square	Emergent (EB)	Solid two-way cast-in-situ slabs (SS2)	Low
Rectangular	Wide (WB)	Solid one-way cast-in-situ slabs (SS1)	Moderate
—	—	Composite slabs with pre-fabricated joists and ceramic blocks (HS)	High

**FIGURE 3** | Illustration of the sampling processes for generating the BCIM data, with sampled data (outputs) highlighted in bold.

the variables describing the building geometry (e.g., by using available probability distributions). The secondary attributes considered herein includes those provided in Table 1, as well as steel and concrete grades, which are mapped to their corresponding material properties during the design stage. Regarding the sampled geometry variables, these are as follows: (1) typical and ground storey heights; (2) standard bay widths along the principal horizontal directions, X and Y ; (3) staircase bay width along the X -axis and (4) layout ID, which can be mapped to a building in-plan configuration in the layouts database. Currently, each configuration in the database is described by the corresponding number of evenly spaced bays in both horizontal directions and the designated staircase location, but the database can be extended to encompass irregular plan layouts. As a result, the BCIM dataset, with the specified sample or portfolio size, comprises information on all the attributes and geometry variables for each building to guide the design process.

Figure 3 illustrates the implemented sampling process, which relies on random generators and decision trees developed based on experience and engineering judgement. The random generators use probability distributions to represent the general characteristics of the building stock, while the decision trees capture correlations between certain random properties and other characteristics, often reflecting assumptions made during the design process. Before beginning the sampling process, for

a given design class, the parameters utilised in the probabilistic models are retrieved from the corresponding data file (a *.json* file named after the design class). While default values are available for each parameter, these can be replaced by alternative user-defined values. This can be particularly useful if more accurate information on the building stock is available. The complete list of parameters that can be input is provided in Tables A1 through A9 of the Appendix A.

After setting the input parameters, discrete probability distributions are first used to assign material grades, quality level, layout ID and column type, each sampled independently. Storey heights are then generated in the following two stages: the typical storey height is drawn from a truncated lognormal distribution, followed by an adjustment factor sampled from a discrete distribution to compute the ground storey height. This value is then capped by a predefined upper limit to ensure feasible configurations. Simultaneously, the staircase bay width along the X -direction is sampled from a uniform distribution over a specified range. Next, standard bay widths in the X - and Y -directions are jointly sampled from a multivariate truncated lognormal distribution, ensuring realistic values while preserving geometric correlations. Once the standard bay widths are established, a decision tree is used to assign the slab type. This process begins by comparing the shortest span length against a predefined threshold; if this threshold is exceeded, the slab type is directly assigned as a HS. If the threshold is not exceeded, the decision logic

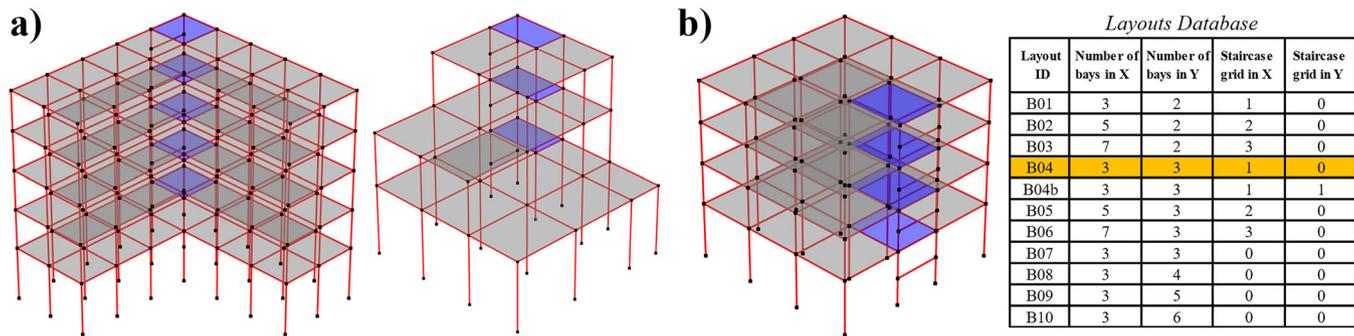


FIGURE 4 | Example of (a) customised and (b) standard frame geometries (rectangles highlighted in blue indicate staircase location whereas those in grey indicate floor slabs).

considers the aspect ratio limits for solid slabs. Based on these criteria, the slab type is then sampled from discrete distributions, resulting in either SS1, SS2 or HS. Following this, a secondary decision step assigns the beam type: if HS is selected, the beam type is sampled from a discrete distribution; otherwise, it is set to EB. This logic ensures that the resulting configurations are both realistic and aligned with typical structural design practices.

Once the sampled data are obtained, the framework instantiates the *StandardGeometry* class that establishes the object representations of the building geometries in Python. For each object (i.e., each building geometry), the grid system is defined based on the specified in-plan configuration and the number of storeys. The grid spacing is then determined using the bay widths and storey heights. Subsequently, lists of simple mesh objects are created to represent different structural components, that is, *Point* for joint, *Line* for beam or column, and *Rectangle* for slab or stairs. All the geometry objects are later used to inform the engine about the connectivity of structural components during the design and numerical modelling stages. As shown in Figure 4, while the framework has the capability to generate irregular geometries, the current layouts database is limited to regular and orthogonal in-plan configurations. Furthermore, the buildings currently considered by the framework feature a single continuous staircase spanning the entire building height. Accordingly, beams spanning along the X-direction and supporting the stairs are included at each mid-storey height to reflect realistic structural behaviour. Once the BCIM and the corresponding *StandardGeometry* objects are generated, they are combined into *TaxonomyData* objects for each building realisation, which serve as the inputs to start the simulated design process that will derive the BDIM.

4 | Building Design Information Model

Regardless of the specific country or region, the seismic design of RC frame buildings typically follows a well-defined sequence of steps, grounded in engineering principles and regulatory building codes. The process usually begins with the definition of seismic loads and general building characteristics. The engineer determines the seismic hazard level for the building's location, for a given return period, often expressed in terms of peak ground acceleration or obtained from an elastic acceleration response spectrum. These parameters are then adjusted

for design purposes, accounting for site-specific characteristics, building importance and the behaviour factors associated with the structural system's ductility. Simultaneously, based on the architectural layouts, the locations of structural members (e.g., beams, columns and slabs) are identified, and the lateral load-resisting system is defined. This is followed by the initial selection of the types of structural members and materials to be used, thus finalising the conceptual design. As mentioned earlier, the framework consolidates this essential information in the *TaxonomyData*, which provides the necessary input to initiate the iterative simulated design procedure.

4.1 | Iterative Design Algorithm

Upon determining the conceptual design of the building and the relevant loads, the initial phase of the iterative design process concerns preliminary member sizing. Initial dimensions for structural members, such as columns, beams and slabs, are selected based on engineering practice rules, minimum dimension requirements specified in building codes and the expected design value of permanent and live design loads (i.e., gravity loads). Basically, a first guess of the member sizes is needed in order to evaluate their suitability to withstand the identified loads. These member sizes are a baseline for subsequent design iterations. As a common practice, once the preliminary sizing is completed, section dimensions are uniformised, along the building height for columns and along continuous spans for beams.

Once the preliminary design is established, the next step involves detailed calculations of the internal forces resulting from each considered loading case. Accordingly, an elastic numerical model of the structure is constructed, and linear elastic analysis is performed for each load case. To better approximate the actual behaviour, stiffness values in the numerical model are adjusted using coefficients to account for cracked sections and not gross section properties, when required by a certain design code. The seismic loads are defined based on the β coefficient and applied using the equivalent lateral force method. However, the framework can also be extended to support the response spectrum analysis method in future implementations. Following this, the resulting member forces are combined by superposition in accordance with the load combinations prescribed by building codes.

In the subsequent step, section dimensions are verified under the given design loads. This verification process involves the following:

1. Assessing the economic feasibility of section dimensions based on engineering rules of thumb;
2. Verifying admissible section stresses or other local checks, if required;
3. Performing global checks, such as drift limits, if mandated by the building code.

In the next step, following either working (or allowable) stress design (for designs based on older codes) or limit state design (for designs based on modern codes), a reinforcement configuration solution is sought for each member section, based on the available steel diameters and reinforcement detailing practices, ensuring all design criteria are satisfied. If capacity design principles are to be followed, the procedure also involves the following:

1. Determining beam longitudinal reinforcement and calculating capacity design shear forces to define the transverse reinforcement;
2. Calculating capacity design bending moments for columns to determine their longitudinal reinforcement, followed by deriving capacity design shear forces to define the transverse reinforcement.

Once reinforcement configurations are determined for each relevant member section, they are uniformised or adjusted to align with construction practices, for example, the same reinforcement configuration is considered at two adjacent beam span ends. Finally, local ductility checks (e.g., verifying maximum longitudinal reinforcement ratios) are performed.

If the design verification fails or no suitable reinforcement configuration is found for a given member section, its dimensions are increased and the steps following preliminary sizing, including the section uniformization, are repeated. This iterative process continues until a successful design solution is achieved. However, if the maximum admissible section dimensions are reached and the design still does not meet the requirements, the initial assumptions defined in the *TaxonomyData* are systematically revised. First, the beam type is checked; if it is defined as a wide beam, it is changed to an emergent beam. If this modification does not lead to a feasible solution, the material properties are revised by selecting higher-strength concrete or steel grades provided such alternatives are available. If no such alternatives exist, the column type is then revised, that is, the type of column is changed to square if it is rectangular. Each time a change is made, whether to beam type, material grade or column shape, the entire design process is restarted, including preliminary sizing and all subsequent steps, considering the new assumptions. If all possible modifications are exhausted and no valid solution is found, the process is terminated without a feasible design solution. This iterative design algorithm was systematically implemented in an automated manner, as illustrated in Figure 5.

4.2 | Design Class Constructors for RC MRF Typologies

In light of the previous discussion, the *bdim* package is structured to reflect the iterative design procedure and serves as a cornerstone of the framework, enabling the simulated design of buildings in compliance with different regional and temporal seismic design practices. Accordingly, *bdim* comprises a series of sub-packages, each named after a specific building design class, thus referred to as Design Class Constructors (DCC). These DCCs implement design methodologies and rules reflecting their respective design codes and regional practices. Additionally, at its core, the package includes a foundational base library, the *baselib* sub-package, which provides a general interface and shared methods for all DCCs, including the implementation of the iterative design procedure, which reflects how structural design follows a set of basic engineering principles, regardless of region or era. As shown in Figure 6, the *baselib* acts as the template from which DCCs inherit or adapt their functionality. It includes abstract classes and core methods that serve as the backbone for all DCCs. Each DCC inherits these base functionalities and overrides or extends them as needed to align with its specific structural design code requirements and/or regional practices. This design architecture allows the *bdim* package to maintain a high degree of modularity and reusability and ensures that new DCCs can be easily integrated into the framework. Developers can focus on implementing code-specific rules for a new design class while leveraging the robust, pre-existing functionalities provided by *baselib*. Since the inherited base classes contain numerous attributes and methods; only the key aspects of each class are highlighted in the following for the sake of brevity.

4.2.1 | Building Class

The framework dynamically maps the *TaxonomyData* to the appropriate building design class implementation, or DCC, generating instances of the corresponding *Building* class. These classes act as a central orchestrator during the simulated design process, consolidating the instances of all other classes and all the methods necessary for executing the iterative design algorithm. In any DCC implementation, if column section dimensions are to be uniformised per a specified number of storeys instead of the entire building height, the relevant step size attribute must be explicitly set. Likewise, if the capacity design principles are to be followed, the safety or overstrength factors associated with capacity design forces—i.e., beam shear, column bending moment and column shear—must be explicitly specified in the relevant attributes of the *Building* classes; otherwise, they will not be included in the design forces. It is worth mentioning that the default methods provided in the *baselib* follow the Eurocode 8 [63] standards for computing these forces. Therefore, if an alternative approach is specified for a new design class, developers must override the corresponding methods to ensure alignment with the relevant specifications. Although not implemented in the current version of the framework, methods for performing global checks, such as verifying prescribed storey drift limits, could be integrated into the *Building* classes if required.

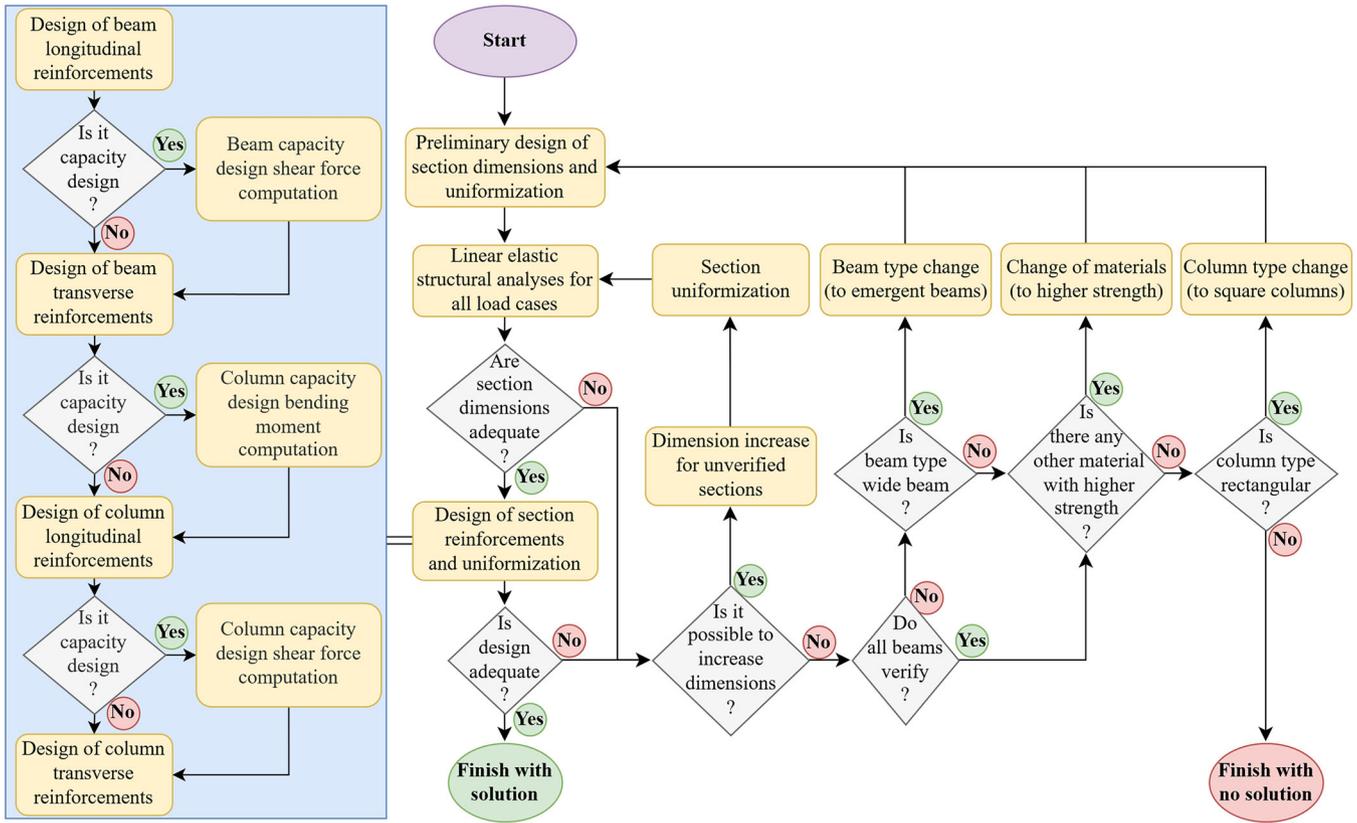


FIGURE 5 | Iterative design algorithm implemented in the framework.

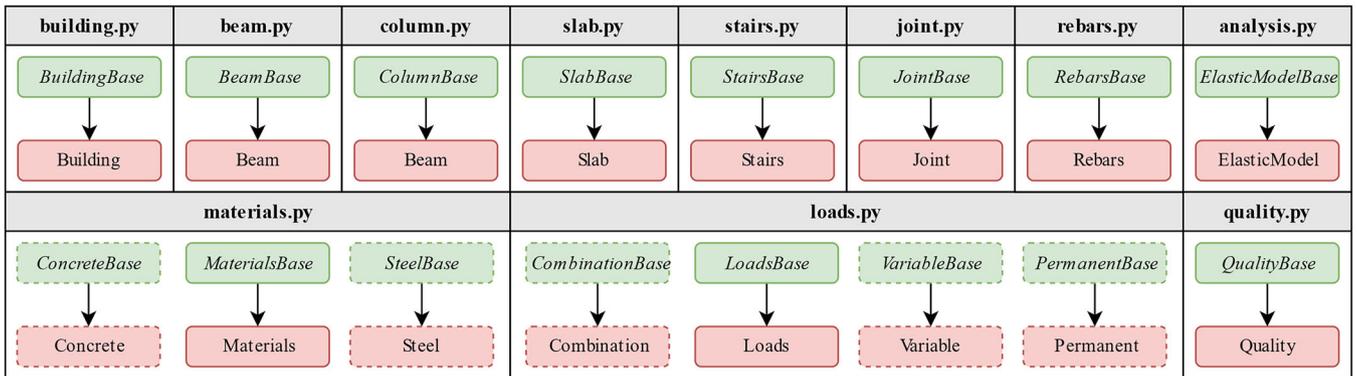


FIGURE 6 | Modules (.py files) within the base design library (*baselib*) and key class inheritances illustrating the relationship between base classes (in green) and design-class-specific implementations (in red).

4.2.2 | Materials Class

Within each *Building* instance, during the design process, material properties needed during the design process are accessed through *Steel* and *Concrete* instances, which encapsulate key characteristics of steel and concrete, such as strength, partial factors, and elastic shear and Young’s moduli. Given that multiple material grades may be relevant for a DCC, each implementation includes a data file (e.g., *materials.json*¹) where specific properties can be explicitly defined and used to create *Steel* and *Concrete* instances for various materials in the database. These instances are managed through a *Materials* object (Figure 6), which allows

the framework to change the basic material properties seamlessly as needed during iterative design processes. Material properties that need to be computed as functions of the basic properties, such as characteristic strengths are derived using default methods provided in the base *Steel* and *Concrete* classes unless explicitly defined in the database. Therefore, when additional material properties are required, they must be explicitly defined in the database. Likewise, if the default formulations for calculating any property differ from the required specifications or practices, the relevant methods should be customised within the DCC to ensure the designs remain compliant with specific codes and practices.

4.2.3 | Loads and Elastic Model Classes

Loading definitions necessary for structural analysis are managed through the instance of the corresponding *Loads* class in any DCC, as shown in Figure 6. For seismic loading, this instance computes the equivalent lateral forces associated with each seismic mass using the β coefficient. As the calculation method may vary between DCCs, it may be necessary to override the corresponding method in the base class. Furthermore, the *Loads* instance integrates *Variable* and *Permanent* objects, which define relevant gravity load sources, and a list of *Combination* objects, providing a comprehensive set of load combinations to represent various design scenarios. To facilitate the generation of these objects, each DCC must include a data file (e.g., *loads.json*²) where the values for variable loads (e.g., live loads on floor, roof and staircase) and permanent loads (e.g., finishes, infill walls and self-weight of RC components) are specified, along with the load cases and corresponding factors for each load combination. Meanwhile, during the design iterations, the *ElasticModel* class is utilised to create linear elastic frame models with rigid diaphragms. It facilitates the structural analyses under various load cases to estimate demands on structural components such as beams and columns for each load combination. For seismic loading, in order to account for cracked stiffness in beams and columns, when required by a certain DCC and regardless of the β value, the effective or reduced moment of inertia is considered in the numerical modelling. Typically, the base implementation of this class does not require modifications in the DCCs, unless additional demand parameters (beyond element forces from load combinations) need to be stored during the analyses.

4.2.4 | Beam and Column Classes

Structural elements such as beams and columns are instantiated as lists of the corresponding *Beam* and *Column* objects, based on the relevant basic geometric representations (i.e., *Line*) described earlier, which are retained as attributes. The *Beam* objects represent rectangular beams, with cross-section dimensions constrained by upper and lower limits, as well as aspect ratios that are defined differently for the two beam types, EB and WB. Each *Beam* object includes a preliminary design method that estimates the initial section dimensions before the iterative design process begins, to ensure a faster convergence towards the final design solution. While a default implementation of this method is provided in the base class, it is highly recommended to override it in new DCCs to address code-specific design criteria or specific economic design practices that want to be considered. In addition to the property methods describing the minimum and maximum section dimensions, *Beam* objects include properties related to reinforcement design, such as admissible minimum and maximum longitudinal reinforcement ratios and minimum transverse reinforcement ratios, which can also be customised in new DCCs. For longitudinal reinforcement, it is assumed that beams include two layers of rebars (top and bottom) at any section, and up to two different bar diameters in each layer. For transverse reinforcement, the stirrup spacing may vary along the beam length. Design forces (i.e., bending moments and shear forces) are calculated at three critical sections along the length of the beam—start, middle and end.

The *Column* objects represent rectangular or square columns, with cross-section dimensions constrained by the upper and lower limits and, in the case of rectangular sections, by the aspect ratio. Like the *Beam* objects, each *Column* object includes a preliminary design method that estimates the initial section dimensions based on a preliminary estimate of the axial load demand, providing a reasonable starting point for the iterative design process. Customising this method in new DCCs is also recommended to incorporate code-specific requirements, such as axial load ratio limits. Similarly, in addition to property methods for maximum and minimum section dimensions, *Column* objects include properties related to the design of the longitudinal and transverse reinforcements. For the longitudinal reinforcement, columns are assumed to have a layer of rebars on each side, symmetrically arranged with respect to the two local axes, where the corner bars can have a different diameter than the internal bars. For the transverse reinforcement, the stirrup spacing can vary along the column and the number of legs on each axis can be different. Design forces (i.e., axial forces, bending moments and shear forces) are calculated at two critical sections along the length of the column—start and end.

The base class defines three abstract methods that must be implemented in new DCCs for both beams and columns to effectively use these design forces in the iterative design processes as follows: verification of section dimension adequacy during the design iterations for the given design loads; calculation of the required longitudinal reinforcement area to ensure sufficient flexural strength and calculation of the required transverse reinforcement ratio per unit spacing to ensure sufficient shear strength. Nevertheless, the default attributes and methods that control the changes to the dimensions during the design iterations and are provided in the base classes can be directly used. Likewise, the methods to compute moment capacities for capacity design calculations are also provided, but they could be changed in specific DCCs if required. It is also worth noting that while the base classes for these objects include the computation of fundamental mechanical properties, the reduction factors considered to compute the effective moment of inertia for these components often vary across DCCs. For example, when considering older design codes, these factors are typically set to one, as cracked section properties were not taken into account.

4.2.5 | Slab, Stairs and Joint Classes

The joints, slabs and stairs are also instantiated as lists of the corresponding objects (*Joint*, *Slab* and *Stairs*) and based on their basic geometric representations (i.e., *Point* and *Rectangle*). For *Slab* objects, their thickness attributes are determined during the initialisation using built-in methods that account for the slab typology and span lengths. These objects are mainly responsible for distributing the gravity loads to the supporting *Beam* objects. In the case of one-way slabs (i.e., SS1 or HS), the loads are transferred along the longer span to two parallel beams, while in the case of two-way slabs, the loads are distributed to the four peripheral beams. The *Stairs* objects serve a similar purpose. Aside from identifying the staircase location and the associated supporting *Beam* objects within the *Building*, *Stairs* objects define the staircase slab thickness and facilitate the transfer of gravity

loads to the supporting beams. Given that design practices for determining the tributary area and the thickness may vary across regions, developers can override these methods for the *Slab* and *Stairs* classes in new DCCs, to ensure the framework aligns with the desired local practices. Meanwhile, the *Joint* objects inform the *Building* class instance about the connectivity of beams and columns at each structural joint. These objects are primarily utilised to compute relevant capacity design forces when necessary and facilitate the numerical modelling later. As their functionality is generally standardised, the need to customise methods within the *Joint* class is unlikely.

4.2.6 | Rebars Class

Reinforcement configurations and detailing for beams and columns are managed by the *Rebars* instance during the design process, utilising the required reinforcement to meet the structural demands computed by the aforementioned methods of the *Beam* and *Column* classes. Furthermore, for each component, the reinforcement configuration is determined by taking into account the available rebar diameters and spacings for longitudinal and transverse reinforcement, as well as the detailing constraints. In each DCC, the diameter and spacing options are retrieved from a data file (e.g., *rebars.json*³), which lists the typical values considered in the corresponding design practice. Meanwhile, the base class provides default methods to define the detailing constraints, such as the maximum spacing between rebars or the minimum rebar diameters. However, it is often necessary to override these methods in each DCC to accommodate differences in building design codes.

4.2.7 | Quality Class

Finally, after the simulated design algorithm produces the design solution, the *Quality* object makes element-wise adjustments to the design properties of beams and columns based on prescribed construction quality levels. More specifically, parameters such as the stirrup spacing, the concrete cover, the mean compressive concrete strength and the mean reinforcing steel yielding strength (for both the longitudinal and the transverse reinforcements) are modified by quality factors to establish expected in situ values that will be adopted in the numerical modelling. The base class provides a default method for randomly sampling these factors for each component using the following predefined probability distributions: a uniform distribution for stirrup spacing and log-normal distributions for other parameters. Accordingly, quality models for the prescribed levels (high, moderate, low) must be introduced for each DCC through a dedicated data file (e.g., *quality.json*⁴), where the distribution parameters for sampling the quality factors are specified for each level. Additionally, the *Quality* object informs the framework about construction quality-related nonlinear numerical modelling aspects utilised in subsequent stages. In particular, each quality model defines the bond-slip factors (ranging from 0 to 1) that are used to define plastic hinge properties in beams and columns, as well as the type of beam-column joint model (rigid, elastic or inelastic). Unless additional design properties need modification or alternative distribution types are required, overriding the methods of the base

class is not expected to be necessary, and providing the referred data file is sufficient to implement quality-related adjustments for any DCC.

5 | Numerical Modelling Development

Considering the quality-adjusted or in situ material properties and reinforcement detailing, the framework transforms the building design data stored in the BDIM into 3D nonlinear structural models, such as the one shown in Figure 7, which can be readily analysed in *OpenSees* [50]. As described earlier, this is accomplished through the *bnsM* package which includes a set of modules designed to manage the structural components effectively. In this package, components such as beams, columns, floors, joints and foundations, along with structural nodes, are represented by classes which encapsulate the essential properties and methods for determining parameters required to model each structural element. Additionally, these classes include methods to export commands for constructing OpenSees objects, as well as the functionality to run these models in real-time. Meanwhile, the *Model* class serves as the orchestrator for assembling the structural model. It streamlines the model construction process, integrating all component-level objects and ensuring compatibility with the design data by incorporating quality-related modelling aspects. Furthermore, the *Model* class includes methods to carry out modal analysis for dynamic characterisation and nonlinear static pushover analysis for evaluating seismic performance. It enables real-time analysis upon constructing the numerical model and supports exporting a comprehensive set of .py and .tcl scripts defining the numerical models and the corresponding analysis routines for OpenSees.

While multiple modelling approaches can be incorporated into the framework, the current implementation adopts a lumped plasticity approach to simulate the nonlinear behaviour of frame elements, as illustrated in Figure 7. Accordingly, at the ends of each frame element, accounting for the rigid joint offsets, *zeroLength* elements are introduced to simulate the moment-rotation response of the expected plastic hinges. For each beam, in-plane flexural behaviour is defined by a single rotational spring whereas, for columns, two rotational springs are defined for each orthogonal direction. More specifically, these springs are defined with the *Hysteretic* uniaxial material model in OpenSees. The yielding moment and the yielding rotation capacity are determined according to Panagiotakos and Fardis [66] and Eurocode 8–Part 3 [29], while the rest of the parameters are obtained according to the Haselton et al. [67] and ASCE/SEI–2017 [68]. The bond-slip factor set by the quality model is included in the computation of the plastic rotation capacity, thus ensuring that construction quality effects are accounted for in the hinge properties. The *zeroLength* elements are connected in-series with a linear elastic interior element where the elastic element stiffness is modified in accordance with Zareian and Medina [69] to prevent unrealistic damping in the model during the dynamic analysis. Although the numerical modelling approach does not account for the interaction between axial flexure behaviours of columns, the properties of the backbone curve of columns are determined considering the axial force corresponding the specified seismic load combination. Unless different load factors are provided,

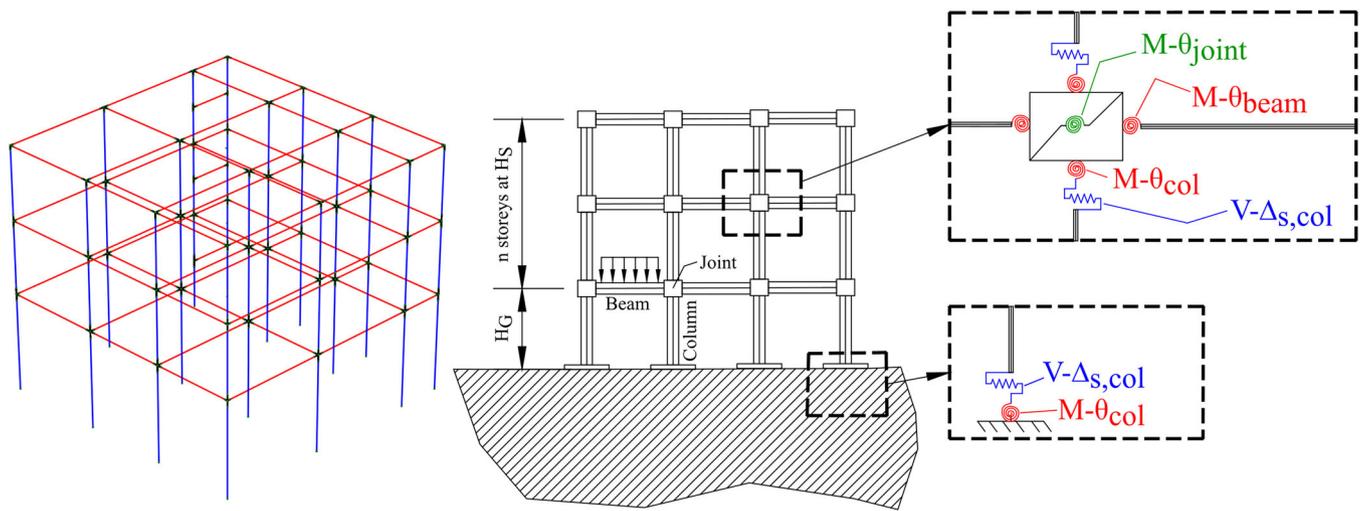


FIGURE 7 | Example of a 3-D model and representation of the numerical modelling approach in a plane-view (H_G , H_S : ground/typical storey heights; $M-\theta$ moment–rotation behaviour of a rotational spring; $V-\Delta_s$: force–displacement behaviour of a shear spring).

this load combination is set by the default factors defined for all permanent and variable loads, respectively. Similarly, these default factors are used for defining gravity loads and masses unless explicitly specified otherwise.

To account for the potential shear failure in columns when capacity design principles are not followed, shear springs are included in the *zeroLength* elements, as represented in Figure 7. The *LimitState* model with the *ThreePoint* limit curve proposed by Elwood and Moehle [70] and Elwood [71] is adopted for the shear hinge material. In particular, the displacement ductility-related shear strength degradation trilinear limit curve proposed by Sezen and Moehle [72] is adopted for each orthogonal direction. Accordingly, the shear strength is obtained using the model proposed in ASCE/SEI–2017 [68], which extends the formulation provided by Sezen and Moehle [72] by incorporating reductions in the truss mechanism strength for sections with widely spaced stirrups. The initial stiffness and the degradation stiffness of the model are obtained using the expressions proposed by LeBorgne and Ghannoum [73] and Shoraka and Elwood [74], respectively.

The beam-column joints shown in Figure 7 are modelled using the *zeroLength* elements placed between the central joint and floor nodes, both defined at the same location. More specifically, the central joint nodes, to which structural masses are assigned, establish the connectivity between beam and column elements, whereas the floor nodes are constrained by the rigid diaphragm to simulate the presence of floor slabs. The joint flexibility is considered only in the rotational degrees-of-freedom with respect to the two horizontal axes. As mentioned earlier, the moment–rotation behaviour of the joints is represented using either inelastic, elastic or rigid materials, depending on the joint type specified in the quality model. The behaviour of inelastic joints is characterised by a *Hysteretic* uniaxial material with material parameters obtained from different expressions specified by O'Reilly [75] and O'Reilly and Sullivan [76] for various joint locations in the building, such as roof, interior or exterior joints. Meanwhile, the stiffness of the elastic joints is derived from the first branches of the backbone curves.

6 | Example Application

To demonstrate the applicability of the framework, several example applications are presented herein. Although, the DCCs inherited from *baselib* are utilised in these examples, their details are not discussed here for the sake of brevity. The objective is to showcase the framework and the possibilities it offers. Future work will look to build on this framework and discuss details regarding the implementation of DCCs for various regions. The complete dataset produced in these examples is available as electronic supplements and at <https://github.com/builtenvdata/simulated-design/tree/main/data/core-article-data>. Still, for conciseness, only a subset of the outputs obtained from the framework is discussed herein.

To showcase building-to-building variability within a building class, an example portfolio consisting of 30 buildings was generated. The considered RC frames were assumed to have four storeys, a design class of CDL, and were designed for a β value of 0.1. To ensure reproducibility, the sampled BCIM data, summarised in Figure 8, were generated using the example values specified in Appendix A regarding the parameters of the probabilistic models. The single-threaded execution for this operation took approximately 123 s on a standard laptop with an 11th Gen Intel Core i7-1165G7 processor, suggesting a reasonable computational performance for large-scale applications. As can be seen from Figure 8, the variability exhibited by both the secondary attributes and geometry variables among the 30 buildings designed using the framework is evident. For instance, the distribution of slab, beam and column types demonstrates a mix of structural configurations. Similarly, the variability in material grades and construction quality levels further reflects differences in construction characteristics of buildings. Geometry variables also exhibit notable variability, including diverse layout IDs numbered according to Figure 4b, storey heights and bay widths (beam span lengths) in both the X and Y directions. Furthermore, the properties such as floor area, floor aspect ratio and total storey height, computed based on the sampled geometry variables, reveal a substantial variation across the portfolio. This highlights the framework's ability to generate a wide

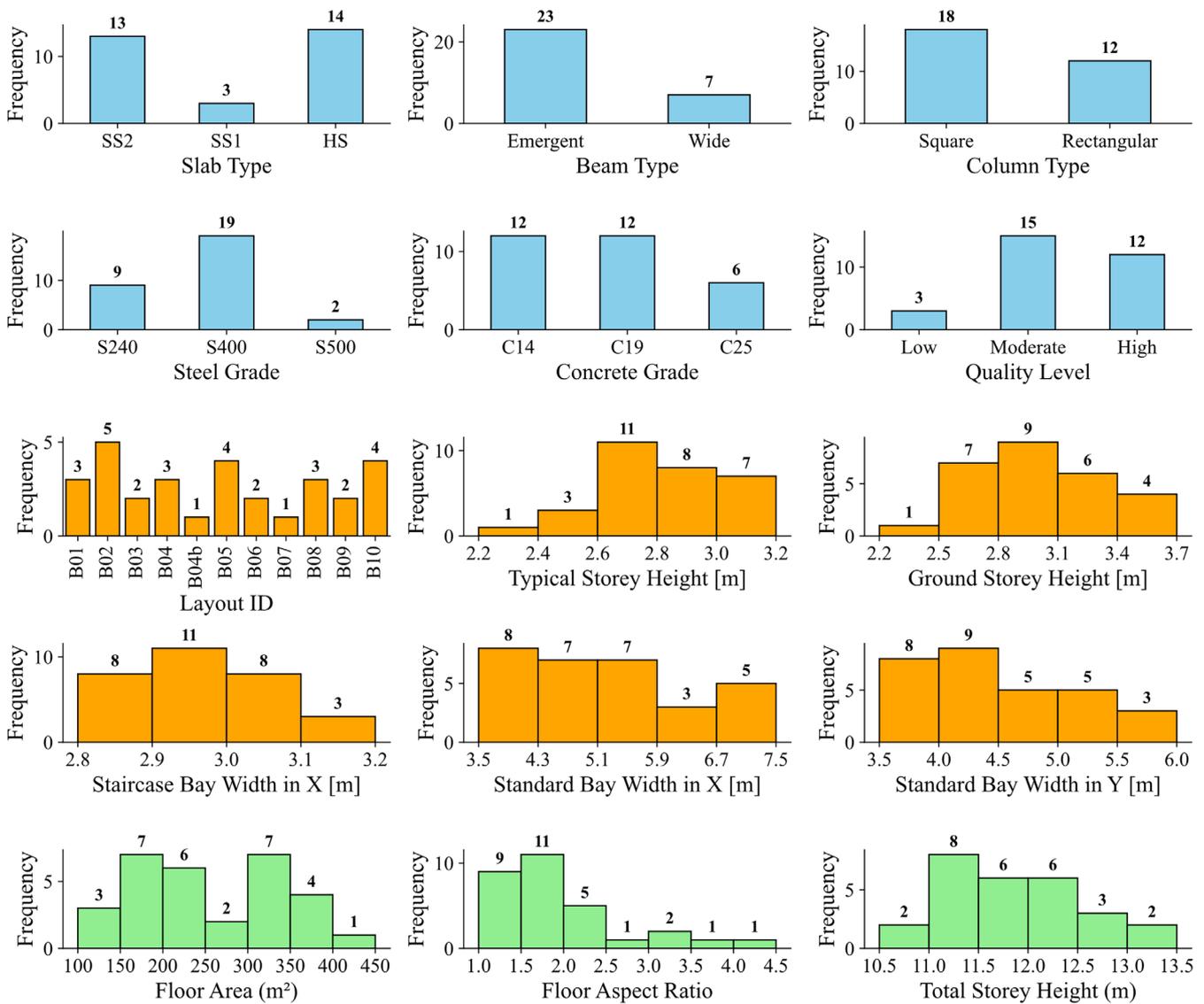


FIGURE 8 | The histograms of generated BCIM data. Light-blue bars represent sampled design attributes, orange bars depict sampled geometrical variables and light-green bars correspond to resulting geometrical properties.

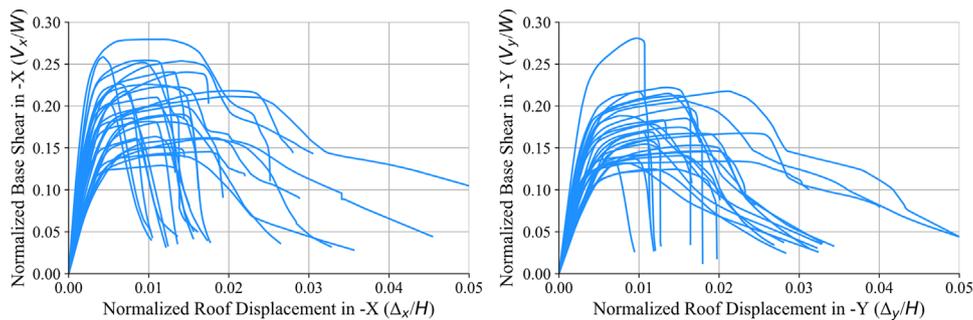


FIGURE 9 | Normalised pushover curves obtained for the generated building nonlinear structural models.

range of realistic configurations consistent with findings in the literature [16, 51, 52, 77–80]. Moreover, the normalised capacity curves obtained from nonlinear static pushover analyses of the simulated building designs under a fundamental mode-based load pattern, as presented in Figure 9, reveal significant differences between the buildings in terms of seismic performance.

While not carried out here, these different design simulations could be analysed using ground motion records to develop fragility and vulnerability models needed for risk assessment. Collectively, these outcomes showcase the framework’s capability to effectively capture building-to-building variability within a portfolio.

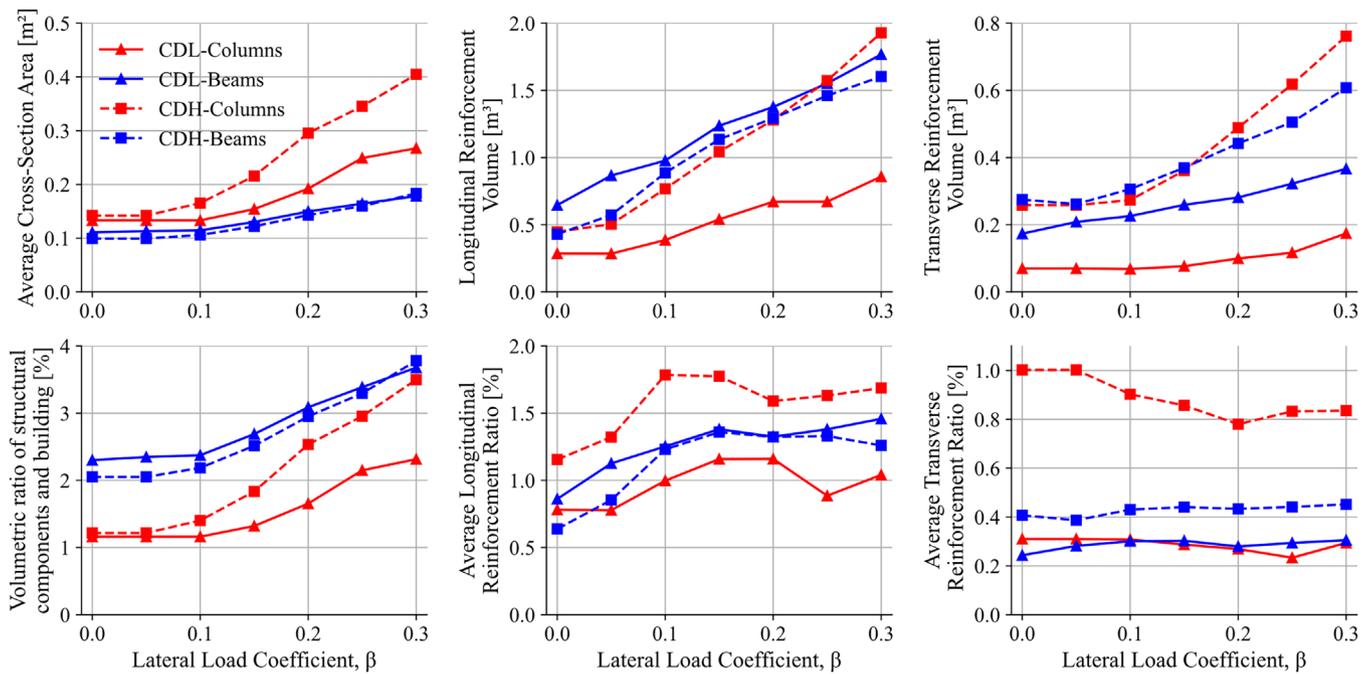


FIGURE 10 | Beam and column properties for CDL and CDH buildings designed for varying β values.

To highlight the simulated design capabilities of the framework, the same BCIM data were processed for two different design classes, CDL and CDH, to generate the corresponding BDIM data at various seismic hazard levels (i.e., different β values). Figure 10 illustrates the resulting beam and column properties across these design classes and seismic hazard levels. The results show a clear trend: both the average section dimensions, which are proportional to concrete volumes, and the total reinforcement volumes increase for higher β values. This outcome is expected since higher seismic hazard levels demand a greater structural capacity to resist larger forces to ensure safety. The increase in concrete and reinforcement volumes highlights how the framework adapts the designs to meet higher seismic demand levels, here represented generally by the design classes CDL and CDH. Additionally, the values shown for the reinforcement ratios and ratios of beam and column volumes to building volume can be seen to be within the ranges reported in the literature [16, 51, 52, 77–80], indicating the realistic nature of the building designs that were obtained.

Figure 10 also highlights an important distinction between the CDL and CDH design classes with respect to transverse reinforcement requirements. CDH beams and columns exhibit significantly higher transverse reinforcement volumes and ratios when compared to those obtained for the components design for the less stringent CDL design class. These differences reflect the stricter requirements of modern design codes, where promoting ductile mechanisms and avoiding brittle failure is a primary concern. Furthermore, in terms of longitudinal reinforcement, CDH columns have substantially higher reinforcement ratios when compared to those of CDL columns, particularly for higher β values. This aligns with the enforcement of capacity design principles inherent to modern design codes [81], which introduce the strong-column and weak-beam concepts to enhance global ductility and reduce the likelihood of soft-storey collapse. Inter-

estingly, the longitudinal reinforcement can be seen to be similar between CDH and CDL beams for higher β levels, while it is higher in CDL beams for lower β levels. In the current implementation of CDL, this difference arises because the gravity load combinations considered for CDL include larger factors at lower hazard levels, resulting in higher reinforcement requirements for beams in these scenarios. However, this deviation from the concept of strong-column to weak-column makes CDL buildings more vulnerable to brittle failure in the event of large seismic events.

To further illustrate the capabilities of the framework, an additional comparative analysis is proposed herein which distinguishes the seismic performance of the different design classes considered by the framework. The BCIM data of the previous example was processed now to determine the BDIM data for one building of each design class designed for varying β values and the same layout. Pushover analyses were then performed using the numerical models of the buildings to determine their normalised capacity curves, which are presented in Figure 11. As can be seen, the CDN curves are the same for all β values since this building class only involves design for gravity loads. On the contrary, the normalised base shear (or strength ratio) values of buildings from the other design classes increase with β , as expected. Moreover, when seismic design is considered ($\beta > 0$), the strength ratios for a given β value progressively increase from CDL to CDH, reflecting the evolution of seismic design. Regarding ductility, CDH buildings exhibit significant ductile behaviour regardless of the β value, as modern capacity design principles are followed in their design. Conversely, buildings of other design classes demonstrate limited levels of ductility. Moreover, as β increases, CDL and CDM buildings begin to exhibit brittle behaviour due to the occurrence of shear or joint failure. The main underlying reason for this behaviour is the lack of capacity design requirements in shear design. As β increases, both

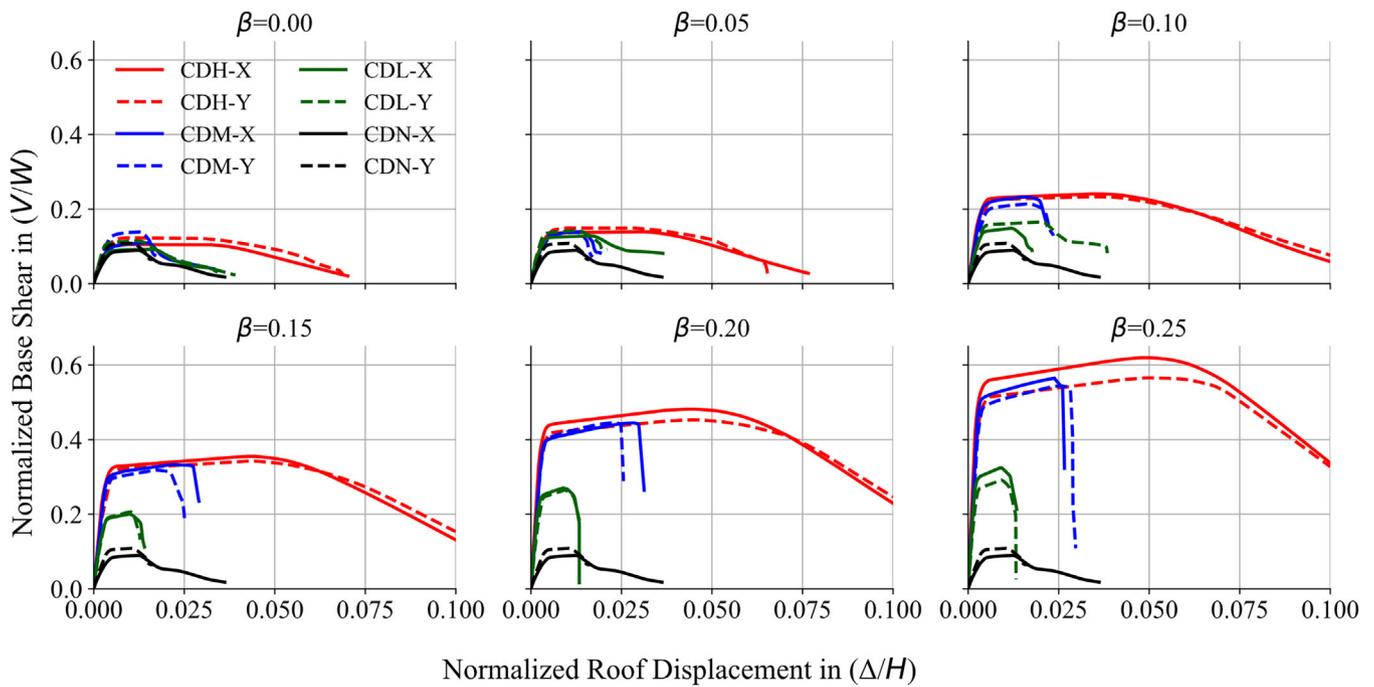


FIGURE 11 | Normalised pushover curves obtained for varying design classes and β values.

the cross-section and longitudinal reinforcement area of beams and columns increase but without guaranteeing the prevention of brittle failure mechanisms. However, it should be noted that brittle failures can also take place for lower β values if different BCIM data are used. Overall, these results underscore critical differences in the structural design philosophies, which can be reflected in the building portfolios obtained by the proposed simulated design process.

7 | Conclusions

This article presented a novel framework for the simulated design of buildings, providing a systematic and adaptable methodology for addressing building-to-building variability and evolving seismic design practices across different regions over time. By combining probabilistic sampling with iterative design algorithms, the framework enables the generation of realistic building designs, and the corresponding numerical models, that are aligned with regional or country-specific contexts. These numerical models, in turn, can facilitate the development of vulnerability models, which reflect the building-to-building variability inherent to the corresponding building class, thereby enhancing the reliability of large-scale (i.e., regional) seismic risk assessments. The framework's open-source Python implementation leverages object-oriented programming principles to ensure modularity and extensibility. This enables the earthquake engineering community to seamlessly integrate the framework into existing workflows, customise it for specific regional applications and extend its capabilities to accommodate different seismic design practices and alternative numerical modelling approaches. The current capabilities of the framework have been demonstrated through a series of examples that emphasise the importance of the taxonomy attributes and geometry variables involved in the design process. The variability of structural

design properties captured in the simulated designs clearly reflects the differences between historical and modern design philosophies. Therefore, these outcomes validate the framework's ability to generate realistic and regionally relevant building portfolios.

The framework was demonstrated here for the specific case of reinforced concrete (RC) moment resisting frames (MRF), and the applications yielded very promising results. Future developments aim to broaden its applicability by incorporating additional structural systems (e.g., RC walls, steel MRFs). Moreover, expanding the geometry database to include irregular layouts and introducing alternative linear elastic analysis methods for both historical and modern seismic design practices, including response spectrum analysis and simpler 2D approaches, would further enhance its versatility and reliability. Similarly, alternative nonlinear modelling approaches to address modelling aspects such as the axial-flexure interaction in columns can be easily introduced given its modular nature. Likewise, since most of the RC frame structures also have the infill walls, which significantly affect the stiffness, strength and overall behaviour of RC structures, they need to be incorporated into the numerical modelling to achieve more accurate representations of building performance. While the framework's adaptability offers substantial opportunities for expanding the range of available design classes through collaborative contributions, this requires further research on the evolution of design codes and practices for individual countries. It is also worth noting that as the framework continues to evolve, it will become possible to validate its outputs through comprehensive comparisons with empirical data and actual design case studies from individual countries. By establishing a community-driven, extensible approach, we anticipate that subsequent research will progressively refine and validate the framework's predictive capabilities and design methodologies. These enhancements will be crucial to demonstrate the accuracy,

reliability and adaptability of the framework across different geographical and regulatory contexts.

In conclusion, the proposed framework not only addresses the current challenges in simulating realistic building designs that are vital when assessing seismic vulnerability and developing research innovations but also lays the groundwork for future advancements through community-driven enhancements. Collaboration within the earthquake engineering community is essential for the continued evolution of this framework. The framework will soon be integrated into the BED initiative through a web-based service and will feature a user-friendly graphical interface. This integration will provide a collaborative platform for advancing seismic design methodologies, promoting shared knowledge and for developing more comprehensive and reliable multi-hazard risk assessments for the built environment.

Acknowledgments

The feedback and comments provided by Helen Crowley and Rui Pinho in the development of this framework are gratefully acknowledged, as is the input of Davit Shahnazaryan on the framework's integration into the BED initiative and Serkan Hasanoglu on its implementation in Python. The authors thank for the financial support from the European Union through the EPOS-ON (Grant Agreement No.101131592) project. The 2nd, 3rd and 4th authors acknowledge the financial support of Funding—UID/04708 of the CONSTRUCT—Instituto de I&D em Estruturas e Construções—funded by Fundação para a Ciência e a Tecnologia, I.P./MCTES through the national funds.

Data Availability Statement

The author has provided the required Data Availability Statement, and if applicable, included functional and accurate links to said data therein.

Endnotes

¹ Available as an electronic supplement and also on the following GitHub repository: https://github.com/builtenvdata/simulated-design/blob/main/simdesign/rcmrf/bdim/eu_cdl/data/materials.json

² Available as an electronic supplement and also on the following GitHub repository: https://github.com/builtenvdata/simulated-design/blob/main/simdesign/rcmrf/bdim/eu_cdl/data/loads.json

³ Available as an electronic supplement and also on the following GitHub repository: https://github.com/builtenvdata/simulated-design/blob/main/simdesign/rcmrf/bdim/eu_cdl/data/rebars.json

⁴ Available as an electronic supplement and also on the following GitHub repository: https://github.com/builtenvdata/simulated-design/blob/main/simdesign/rcmrf/bdim/eu_cdl/data/quality.json

References

1. L. Martins and V. Silva, “Development of a Fragility and Vulnerability Model for Global Seismic Risk Analyses,” *Bulletin of Earthquake Engineering* 19, no. 15 (2021): 6719–6745, <https://doi.org/10.1007/s10518-020-00885-1>.

2. G. M. Calvi, R. Pinho, G. Magenes, J. J. Bommer, L. F. Restrepo-Vélez, and H. Crowley, “Development of Seismic Vulnerability Assessment Methodologies Over the Past 30 Years,” *ISET Journal of Earthquake Technology* 43, no. 3 (2006): 75–104.

3. G. M. Calvi, “A Displacement-Based Approach for Vulnerability Evaluation of Classes of Buildings,” *Journal of Earthquake Engineering* 3, no. 3 (1999): 411–438, <https://doi.org/10.1080/13632469909350353>.

4. H. Crowley and R. Pinho, “Period-Height Relationship for Existing European Reinforced Concrete Buildings,” *Journal of Earthquake Engineering* 8 (2004): 93–119, <https://doi.org/10.1080/13632460409350522>.

5. A. M. B. Nafeh and G. J. O'Reilly, “Fragility Functions for Non-Ductile Infilled Reinforced Concrete Buildings Using Next-Generation Intensity Measures Based on Analytical Models and Empirical Data From Past Earthquakes,” *Bulletin of Earthquake Engineering* 22, no. 10 (2024): 4983–5021, <https://doi.org/10.1007/S10518-024-01955-4>.

6. FEMA, *HAZUS-MH MR5, Technical Manual* (Department of Homeland Security—Federal Emergency Management Agency, 2014).

7. P. Mouroux and B. Le Brun, “Presentation of RISK-UE Project,” *Bulletin of Earthquake Engineering* 4, no. 4 (2006): 323–339, <https://doi.org/10.1007/S10518-006-9020-3>.

8. G. M. Calvi and R. Pinho, *LESSLOSS: A European Integrated Project on Risk Mitigation for Earthquakes and Landslides* (2004).

9. M. Villar-Vega, V. Silva, H. Crowley, et al., “Development of a Fragility Model for the Residential Building Stock in South America,” *Earthquake Spectra* 33, no. 2 (2017): 581–604, <https://doi.org/10.1193/010716EQS005M>.

10. M. A. Erberik, “Fragility-Based Assessment of Typical Mid-Rise and Low-Rise RC Buildings in Turkey,” *Engineering Structures* 30, no. 5 (2008): 1360–1374, <https://doi.org/10.1016/J.ENGSTRUCT.2007.07.016>.

11. M. A. Erberik, “Generation of Fragility Curves for Turkish Masonry Buildings Considering In-Plane Failure Modes,” *Earthquake Engineering & Structural Dynamics* 37, no. 3 (2008): 387–405, <https://doi.org/10.1002/EQE.760>.

12. M. Erdik, N. Aydinoglu, Y. Fahjan, et al., “Earthquake Risk Assessment for Istanbul Metropolitan Area,” *Earthquake Engineering and Engineering Vibration* 2, no. 1 (2003): 1–23, <https://doi.org/10.1007/BF02857534>.

13. B. Borzi, H. Crowley, and R. Pinho, “Simplified Pushover-Based Earthquake Loss Assessment (SP-BELA) Method for Masonry Buildings,” *International Journal of Architectural Heritage* 2, no. 4 (2008): 353–376, <https://doi.org/10.1080/15583050701828178>.

14. B. Borzi, R. Pinho, and H. Crowley, “Simplified Pushover-Based Vulnerability Analysis for Large-Scale Assessment of RC Buildings,” *Engineering Structures* 30, no. 3 (2008): 804–820, <https://doi.org/10.1016/j.engstruct.2007.05.021>.

15. M. A. Salgado, D. Zuloaga, G. A. Bernal, M. G. Mora, and O. D Cardona, “Probabilistic Seismic Risk Assessment of the Building Stock in Medellín, Colombia,” in *Computational Civil Engineering 2013, International Symposium* (2013): 120–135.

16. V. Silva, H. Crowley, H. Varum, R. Pinho, and L. Sousa, “Investigation of the Characteristics of Portuguese Regular Moment-Frame RC Buildings and Development of a Vulnerability Model,” *Bulletin of Earthquake Engineering* 13, no. 5 (2015): 1455–1490, <https://doi.org/10.1007/s10518-014-9669-y>.

17. V. Silva, H. Crowley, H. Varum, and R. Pinho, “Seismic Risk Assessment for Mainland Portugal,” *Bulletin of Earthquake Engineering* 13, no. 2 (2015): 429–457, <https://doi.org/10.1007/s10518-014-9630-0>.

18. H. Motamed, A. Calderon, V. Silva, and C. Costa, “Development of a Probabilistic Earthquake Loss Model for Iran,” *Bulletin of Earthquake Engineering* 17, no. 4 (2019): 1795–1823, <https://doi.org/10.1007/s10518-018-0515-5>.

19. V. Silva, S. Akkar, J. Baker, et al., “Current Challenges and Future Trends in Analytical Fragility and Vulnerability Modeling,” *Earthquake Spectra* 35, no. 4 (2019): 1927–1952, <https://doi.org/10.1193/042418EQS1010>.

20. H. Crowley, V. Despotaki, D. Rodrigues, et al., “Exposure Model for European Seismic Risk Assessment,” *Earthquake Spectra* 36, no. 1 (2020): 252–273, <https://doi.org/10.1177/8755293020919429>.

21. Eurostat, Business Economy by Sector 2025, accessed March 30, 2025, https://ec.europa.eu/eurostat/statistics-explained/index.php?title=Business_economy_by_sector_-_NACE_Rev_2.
22. ISTAT, Italian National Institute of Statistics: Census Data, <https://www.istat.it/it/censimenti/popolazione-e-abitazioni>.
23. FEMA, *Rapid Visual Screening of Buildings for Potential Seismic Hazards: A Handbook* (Government Printing Office, 2017).
24. C. Ratti and P. Richens, "Urban Texture Analysis With Image Processing Techniques," in *Computers in Building: Proceedings of the CAADfutures'99 Conference. Proceedings of the Eighth International Conference on Computer Aided Architectural Design Futures held at Georgia Institute of Technology, Atlanta, Georgia, USA on June 7-8, 1999* (1999): 49-64, https://doi.org/10.1007/978-1-4615-5047-1_4.
25. P. Ricci, G. C. Del, G. M. Verderame, G. Manfredi, M. Pollino, and F. Borfecchia, "Seismic Vulnerability Assessment at Urban Scale Based on Different Building Stock Data Sources," in *Vulnerability, Uncertainty, and Risk: Quantification, Mitigation, and Management* (2014): 1027-1038, <https://doi.org/10.1061/9780784413609.104>.
26. K. Pitilakis, H. Crowley, and A. M. Kaynia, "SYNER-G: Typology Definition and Fragility Functions for Physical Elements at Seismic Risk: Buildings, Lifelines, Transportation Networks and Critical Facilities," *Geotechnical, Geological and Earthquake Engineering* 27 (2014): 1-28, <https://doi.org/10.1007/978-94-007-7872-6>.
27. S. Brzev, C. Scawthorn, A. W. Charleson, et al., *GEM Building Taxonomy Version 2.0.*, vol. 02 (GEM Foundation, 2013), <https://doi.org/10.13117/GEM.EXP-MOD.TR2013.02>.
28. H. Crowley, V. Despotaki, V. Silva, et al., "Model of Seismic Design Lateral Force Levels for the Existing Reinforced Concrete European Building Stock," *Bulletin of Earthquake Engineering* 19, no. 7 (2021): 2839-2865, <https://doi.org/10.1007/s10518-021-01083-3>.
29. CEN, *Eurocode 8: Design of Structures for Earthquake Resistance—Part 3: Assessment and Retrofitting of Buildings* (2005).
30. G. M. Verderame, M. Polese, C. Mariniello, and G. Manfredi, "A Simulated Design Procedure for the Assessment of Seismic Capacity of Existing Reinforced Concrete Buildings," *Advances in Engineering Software* 41, no. 2 (2010): 323-335, <https://doi.org/10.1016/j.advengsoft.2009.06.011>.
31. G. J. O'Reilly and T. J. Sullivan, "Probabilistic Seismic Assessment and Retrofit Considerations for Italian RC Frame Buildings," *Bulletin of Earthquake Engineering* 16, no. 3 (2018): 1447-1485, <https://doi.org/10.1007/S10518-017-0257-9>.
32. S. Ruggieri, M. Calò, A. Cardelicchio, and G. Uva, "Analytical-Mechanical Based Framework for Seismic Overall Fragility Analysis of Existing RC Buildings in Town Compartments," *Bulletin of Earthquake Engineering* 20 (2022): 8179-8216, <https://doi.org/10.1007/s10518-022-01516-7>.
33. M. Polese, M. Marcolini, G. Zuccaro, and F. Cacace, "Mechanism Based Assessment of Damage-Dependent Fragility Curves for RC Building Classes," *Bulletin of Earthquake Engineering* 13, no. 5 (2015): 1323-1345, <https://doi.org/10.1007/s10518-014-9663-4>.
34. A. Masi, "Seismic Vulnerability Assessment of Gravity Load Designed R/C Frames," *Bulletin of Earthquake Engineering* 1, no. 3 (2003): 371-395, <https://doi.org/10.1023/B:BEEE.0000021426.31223.60>.
35. A. Masi and M. Vona, "Vulnerability Assessment of Gravity-Load Designed RC Buildings: Evaluation of Seismic Capacity Through Non-Linear Dynamic Analyses," *Engineering Structures* 45 (2012): 257-269, <https://doi.org/10.1016/j.engstruct.2012.06.043>.
36. A. Masi, S. Lagomarsino, M. Dolce, V. Manfredi, and D. Ottonelli, "Towards the Updated Italian Seismic Risk Assessment: Exposure and Vulnerability Modelling," *Bulletin of Earthquake Engineering* 19, no. 8 (2021): 3253-3286, <https://doi.org/10.1007/s10518-021-01065-5>.
37. C. Del Gaudio, P. Ricci, G. M. Verderame, and G. Manfredi, "Development and Urban-Scale Application of a Simplified Method for Seismic Fragility Assessment of RC Buildings," *Engineering Structures* 91 (2015): 40-57, <https://doi.org/10.1016/j.engstruct.2015.01.031>.
38. L. Martins, V. Silva, P. Bazzurro, and M. Marques, "Advances in the Derivation of Fragility Functions for the Development of Risk-Targeted Hazard Maps," *Engineering Structures* 173 (2018): 669-680, <https://doi.org/10.1016/J.ENGSTRUCT.2018.07.028>.
39. A. M. B. Nafeh and G. J. O'Reilly, "Unbiased Simplified Seismic Fragility Estimation of Non-Ductile Infilled RC Structures," *Soil Dynamics and Earthquake Engineering* 157 (2022): 107253, <https://doi.org/10.1016/J.SOILDYN.2022.107253>.
40. H. M. Crowley, J. Dabbeek, V. Despotaki, et al., "European Seismic Risk Model (ESRM20)," in *EFEHR Technical Report 002 V1.0.0* (2021), <https://doi.org/10.7414/EUC-EFEHR-TR002-ESRM20>.
41. F. Pavel, H. Crowley, X. Romão, R. Vladut, N. Pereira, and V. Ozsarac, "Evolution of National Codes for the Design of RC Structures in Romania," *Bulletin of Earthquake Engineering* 22, no. 3 (2024): 911-949, <https://doi.org/10.1007/s10518-023-01791-y>.
42. G. M. Calvi, G. Magenes, and S. Pampanin, "Relevance of Beam-Column Joint Damage and Collapse in RC Frame Assessment," *Journal of Earthquake Engineering* 6, no. 1 (2002): 75-100, <https://doi.org/10.1142/S1363246902000693>. SPEC. ISS.
43. S. Pampanin, G. M. Calvi, and M. Moratti, "Seismic Behaviour of R.C. Beam-Column Joints Designed for Gravity Loads," *12th European Conference on Earthquake Engineering* 726 (2002): 1-10.
44. H. Sezen, A. S. Whittaker, K. J. Elwood, and K. M. Mosalam, "Performance of Reinforced Concrete Buildings During the August 17, 1999 Kocaeli, Turkey Earthquake, and Seismic Design and Construction Practise in Turkey," *Engineering Structures* 25, no. 1 (2003): 103-114, [https://doi.org/10.1016/S0141-0296\(02\)00121-9](https://doi.org/10.1016/S0141-0296(02)00121-9).
45. E. Vuran, C. Serhatoğlu, M. Ö. Timurağaoğlu, E. Smyrou, İ. E. Bal, and R. Livaoğlu, "Damage Observations of RC Buildings From 2023 Kahramanmaraş Earthquake Sequence and Discussion on the Seismic Code Regulations," *Bulletin of Earthquake Engineering* 23, no. 3 (2025): 1153-1182, <https://doi.org/10.1007/S10518-023-01843-3>.
46. A. Yakut, H. Sucuoğlu, B. Binici, et al., "Performance of Structures in İzmir After the Samos Island Earthquake," *Bulletin of Earthquake Engineering* 20, no. 14 (2022): 7793-7818, <https://doi.org/10.1007/S10518-021-01226-6>.
47. P. Ricci, F. de Luca, and G. M. Verderame, "6th April 2009 L'Aquila Earthquake, Italy: Reinforced Concrete Building Performance," *Bulletin of Earthquake Engineering* 9, no. 1 (2011): 285-305, <https://doi.org/10.1007/S10518-010-9204-8>.
48. D. Shahnazaryan, R. Pinho, H. Crowley, and G. J. O'Reilly, "The Built Environment Data Platform for Experimental Test Data in Earthquake Engineering," *Earthquake Engineering & Structural Dynamics* 53, no. 15 (2024): 4627-4640, <https://doi.org/10.1002/EQE.4231>.
49. M. Caruso, V. Silva, K. Aljawhari, A. M. B. Nafeh, and C. Galasso, "Unveiling the Environmental Impact of Earthquakes in Europe (2024)," <https://doi.org/10.21203/RS.3.RS-5283610/V1>.
50. F. McKenna, M. H. Scott, and G. L. Fenves, "Nonlinear Finite-Element Analysis Software Architecture Using Object Composition," *Journal of Computing in Civil Engineering* 24, no. 1 (2010): 95-107, [https://doi.org/10.1061/\(asce\)jcp.1943-5487.00000002](https://doi.org/10.1061/(asce)jcp.1943-5487.00000002).
51. I. E. Bal, H. Crowley, R. Pinho, and F. G. Gülay, "Detailed Assessment of Structural Characteristics of Turkish RC Building Stock for Loss Assessment Models," *Soil Dynamics and Earthquake Engineering* 28, no. 10-11 (2008): 914-932, <https://doi.org/10.1016/j.soildyn.2007.10.005>.
52. C. Del Gaudio, P. Ricci, G. M. Verderame, and G. Manfredi, "Development and Urban-Scale Application of a Simplified Method for Seismic Fragility Assessment of RC Buildings," *Engineering Structures* 91 (2015): 40-57, <https://doi.org/10.1016/j.engstruct.2015.01.031>.

53. M. Zhu, F. McKenna, and M. H. Scott, "OpenSeesPy: Python Library for the OpenSees Finite Element Framework," *SoftwareX* 7 (2018): 6–11, <https://doi.org/10.1016/j.softx.2017.10.009>.
54. G. van Rossum, *Python Tutorial* (1995).
55. RBA, *Regulamento para o Emprego de Betão Armado* (1935).
56. REBA, *Regulamento de Estruturas de Betão Armado* (1967).
57. A. Guerrin, *Traité De Béton Armé* (Dunod, 1966).
58. Comité Européen du Béton (CEB), *Recommandations Pratiques à l'Usage des Constructeurs* (Comité Européen du Béton, 1963).
59. REBAP, *Regulamento de Estruturas de Betão Armado e PréEsforçado* (1983).
60. J. d'Arga e Lima, V. Monteiro, and M. Mun, *Betão Armado—Esforços Normais e de flexão (REBAP - 83)* (Laboratório Nacional de Engenharia Civil, 2005).
61. Comité Européen du Béton (CEB), *Système International de Réglementation Technique Unifiée des Structures, Volume 1 - règles unifiées communes aux différents types d'ouvrages et de matériaux* (Comité Européen du Béton, 1978).
62. Comité Européen du Béton (CEB), *Système International de Réglementation Technique Unifiée des Structures, Volume 2 - Code-modèle CEB-FIP pour les structures en béton* (Comité Européen du Béton, 1978).
63. CEN, Eurocode 8: Design of Structures for Earthquake Resistance—Part 1: General Rules, Seismic Actions and Rules for Buildings (2004).
64. CEN, Eurocode 2: Design of Concrete Structures—Part 1-1: General Rules and Rules for Buildings (2004).
65. X. Romão, J. M. Castro, N. Pereira, et al., "Deliverable Report D26.5 European Physical Vulnerability Models," in *SERA—Seismology and Earthquake Engineering Research Infrastructure Alliance for Europe*, vol. 26 (2019).
66. T. B. Panagiotakos and M. N. Fardis, "Deformations of Reinforced Concrete Members at Yielding and Ultimate," *ACI Structural Journal* 98, no. 2 (2001): 135–148, <https://doi.org/10.14359/10181>.
67. C. B. Haselton, A. B. Liel, S. C. Taylor-Lange, and G. G. Deierlein, "Calibration of Model to Simulate Response of Reinforced Concrete Beam-Columns to Collapse," *ACI Structural Journal* 113, no. 6 (2016): 1141–1152, <https://doi.org/10.14359/51689245>.
68. ASCE, *Seismic Evaluation and Retrofit of Existing Buildings (ASCE/SEI 41-17)*, vol. 41 (American Society of Civil Engineers, 2017).
69. F. Zareian and R. A. Medina, "A Practical Method for Proper Modeling of Structural Damping in Inelastic Plane Structural Systems," *Computers and Structures* 88, no. 1–2 (2010): 45–53, <https://doi.org/10.1016/j.compstruc.2009.08.001>.
70. K. J. Elwood and J. P. Moehle, *Shake Table Tests and Analytical Studies on the Gravity Load Collapse of Reinforced Concrete Frames*, PEER Report 2003/01 (Pacific Earthquake Engineering Research Center, College of Engineering, University of California, 2003).
71. K. J. Elwood, "Modelling Failures in Existing Reinforced Concrete Columns," *Canadian Journal of Civil Engineering* 31, no. 5 (2004): 846–859, <https://doi.org/10.1139/L04-040>.
72. H. Sezen and J. P. Moehle, "Shear Strength Model for Lightly Reinforced Concrete Columns," *Journal of Structural Engineering* 130, no. 11 (2004): 1692–1703, [https://doi.org/10.1061/\(asce\)0733-9445\(2004\)130:11\(1692\)](https://doi.org/10.1061/(asce)0733-9445(2004)130:11(1692)).
73. M. R. LeBorgne and W. M. Ghannoum, "Calibrated Analytical Element for Lateral-Strength Degradation of Reinforced Concrete Columns," *Engineering Structures* 81 (2014): 35–48, <https://doi.org/10.1016/j.engstruct.2014.09.030>.
74. M. B. Shoraka and K. J. Elwood, "Mechanical Model for Non Ductile Reinforced Concrete Columns," *Journal of Earthquake Engineering* 17, no. 7 (2013): 937–957, <https://doi.org/10.1080/13632469.2013.794718>.
75. G. J. O'reilly, Performance-Based Seismic Assessment and Retrofit of Existing RC Frame Buildings in Italy (Scuola Universitaria Superiore IUSS Pavia, 2016).
76. G. J. O'Reilly and T. J. Sullivan, "Modeling Techniques for the Seismic Assessment of the Existing Italian RC Frame Structures," *Journal of Earthquake Engineering* 23, no. 8 (2019): 1262–1296, <https://doi.org/10.1080/13632469.2017.1360224>.
77. T. E. Azak, B. Ö. Ay, and S Akkar, "A Statistical Study on Geometrical Properties of Turkish Reinforced Concrete Building Stock," in *Second European Conference on Earthquake Engineering and Seismology* (2014), 1–11.
78. H. B. Ozmen, M. Inel, S. M. Senel, and A. H. Kayhan, "Load Carrying System Characteristics of Existing Turkish Rc Building Stock," *International Journal of Civil Engineering* 13, no. 1 (2015): 76–91.
79. E. Meral, "Evaluation of Structural Properties of Existing Turkish RC Building Stock," *Iranian Journal of Science and Technology—Transactions of Civil Engineering* 43, no. 3 (2019): 445–462, <https://doi.org/10.1007/s40996-018-0207-z>.
80. A. Furtado, C. Costa, A. Arêde, and H. Rodrigues, "Geometric Characterisation of Portuguese RC Buildings With Masonry Infill Walls," *European Journal of Environmental and Civil Engineering* 20, no. 4 (2016): 396–411, <https://doi.org/10.1080/19648189.2015.1039660>.
81. M. N. Fardis, "Capacity Design: Early History," *Earthquake Engineering and Structural Dynamics* 47, no. 14 (2018): 2887–2896, <https://doi.org/10.1002/eqe.3110>.

Supporting Information

Additional supporting information can be found online in the Supporting Information section.

Appendix

TABLE A1 | The complete list of input parameters utilised in BCIM generation.

Parameter (type)	Description	Example input
*design_class (str)	Seismic design class of the building	“eu_cdl”
*sample_size (int)	Size of the sample to be generated	30
*beta (float)	Design lateral load coefficient	0.1
*num_storeys (int)	Number of storeys in the building, $x \in Z \cap [1,8]$	4
seed (int)	Seed value for random number generation	2
square_column_prob (float)	Probability of having square columns, $x \in R \cap [0.0,1.0]$	0.5
layout (list[str])	List of layout ids considered for building generation (equal probability is assigned to each)	[“B01”, ..., “B10”]
wb_prob_given_hs (float)	Probability of having wide beams (WB) given slab type is HS, $x \in R \cap [0.0,1.0]$	0.5
typical_storey_height (dict)	Dictionary containing the nested parameters required for typical storey height sampling	See Table A2 for nested parameters
ground_storey_height (dict)	Dictionary containing the parameters required for ground storey height sampling	See Table A3 for nested parameters
steel (dict)	Dictionary containing the parameters required for steel grade sampling	See Table A4 for nested parameters
concrete (dict)	Dictionary containing the parameters required for concrete grade sampling	See Table A5 for nested parameters
construction_quality (dict)	Dictionary containing the parameters required for sampling construction quality levels	See Table A6 for nested parameters
staircase_bay_width (dict)	Dictionary containing the parameters required for sampling staircase bay widths	See Table A7 for nested parameters
standard_bay_width (dict)	Dictionary containing the parameters required for sampling standard bay widths in principle horizontal directions X and Y	See Table A8 for nested parameters
slab_typology (dict)	Dictionary containing the parameters required for slab typology sampling	See Table A9 for nested parameters

*Required.

TABLE A2 | The nested parameters of ‘typical_storey_height’.

Parameter (type)	Description	Example input
cv (float)	Coefficient of variation (standard deviation/mean)	0.07
mu (float)	Mean value for the typical storey heights	2.9
lower_bound (float)	Lower bound value for sampling	2.3
upper_bound (float)	Upper bound value for sampling	3.8

TABLE A3 | The nested parameters of ‘ground_storey_height’.

Parameter (type)	Description	Example input
maximum (float)	Maximum possible ground storey height	4.2
factor (list[float])	List of adjustment factors for ground storey heights	[1.0, 1.1, 1.2, 1.3, 1.4]
Probability (list[float])	List of probabilities for the occurrence of each factor (summation should be equal to 1.0)	[0.55, 0.10, 0.20, 0.10, 0.05]

TABLE A4 | The nested parameters of 'steel'.

Parameter (type)	Description	Example input
grade (<i>list[str]</i>)	List of steel grades or tags used in the dataset	["S240", "S400", "S500"]
Probability (<i>list[float]</i>)	List of probabilities for the occurrence of each steel grade (summation should be equal to 1.0)	[0.20, 0.70, 0.10]

TABLE A5 | The nested parameters of 'concrete'.

Parameter (type)	Description	Example input
grade (<i>list[str]</i>)	List of concrete grades or tags used in the dataset	["C14", "C19", "C25"]
probability (<i>list[float]</i>)	List of probabilities for the occurrence of each concrete grade (summation should be equal to 1.0)	[0.40, 0.40, 0.20]

TABLE A6 | The nested parameters of 'construction_quality'.

Parameter (type)	Description	Example input
quality (<i>list[int]</i>)	List of quality levels represented as IDs, $x \in \{1,2,3\}$ (1: High, 2: Moderate, 3: Low)	[1, 2, 3]
probability (<i>list[float]</i>)	List of probabilities for the occurrence of each quality level (summation should be equal to 1.0)	[0.60, 0.30, 0.10]

TABLE A7 | The nested parameters of 'staircase_bay_width'.

Parameter (type)	Description	Example input
lower_bound (<i>float</i>)	Minimum possible width of the staircase bay	2.8
upper_bound (<i>float</i>)	Maximum possible width of the staircase bay	3.2

TABLE A8 | The nested parameters of 'standard_bay_width'.

Parameter (type)	Description	Example input
corr_coeff_xy (<i>float</i>)	Correlation coefficient between X and Y direction bay widths, $x \in \mathbb{R} \cap [-1.0, 1.0]$	-0.92
lower_bound_x (<i>float</i>)	Lower bound of truncated log-normal distribution for X direction	3.5
upper_bound_x (<i>float</i>)	Upper bound of truncated log-normal distribution for X direction	7.5
theta_x (<i>float</i>)	Median of log-normal distribution for X direction	4.5
sigma_x (<i>float</i>)	Logarithmic standard deviation for X direction	0.35
lower_bound_y (<i>float</i>)	Lower bound of truncated log-normal distribution for Y direction	3.5
upper_bound_y (<i>float</i>)	Upper bound of truncated log-normal distribution for Y direction	7.5
theta_y (<i>float</i>)	Median of log-normal distribution for Y direction	4.5
sigma_y (<i>float</i>)	Logarithmic standard deviation for Y direction	0.35

TABLE A9 | The nested parameters of 'slab_typology'.

Parameter (type)	Description	Example Input
ss1_prob_given_ss1_or_hs (<i>float</i>)	Probability of SS1 type slab given SS1 or HS, $x \in \mathbb{R} \cap [0.0, 1.0]$	0.5
ss2_prob_given_ss2_or_hs (<i>float</i>)	Probability of SS2 type slab given SS2 or HS, $x \in \mathbb{R} \cap [0.0, 1.0]$	0.65
max_ss_short_span (<i>float</i>)	Upper limit for the short span length in solid slabs (SS1, SS2), in metres	6.0
max_ss2_aspect_ratio (<i>float</i>)	Upper limit for the aspect ratio in SS2 slabs $x \in \mathbb{R} \cap (1.0, \infty)$	2.0